

[illegible]

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```

1  G001 0 MODULE QUEMANSHO(%TITLE 'SHOW QUEUE command'
2  0002 0 IDENT = 'V04-000'
3  0003 0 ) =
4  0004 1 BEGIN
5  0005 1
6  0006 1
7  0007 1 *****
8  0008 1 *
9  0009 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 *   ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 *   TRANSFERRED.
19 0019 1 *
20 0020 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 *   CORPORATION.
23 0023 1 *
24 0024 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1   Queue management commands.
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1   This module implements the SHOW QUEUE command.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1   VAX/VMS user mode.
40 0040 1 --
41 0041 1
42 0042 1 AUTHOR: M. Jack, CREATION DATE: 10-Nov-1982
43 0043 1
44 0044 1 MODIFIED BY:
45 0045 1
46 0046 1   V03-013 JAK0225      J A Krycka      16-Aug-1984
47 0047 1   Restrict display of DEFAULT, SEPARATE, and FORM attributes
48 0048 1   to nongeneric printer and terminal queues.
49 0049 1
50 0050 1   V03-012 JAK0222      J A Krycka      06-Aug-1984
51 0051 1   Fix bug in display of DEFAULT list introduced in JAK0220.
52 0052 1
53 0053 1   V03-011 JAK0220      J A Krycka      20-Jul-1984
54 0054 1   Support display of /DEFAULT=FEED and [NO]FEED.
55 0055 1
56 0056 1   V03-010 PCG0003      Peter George    13-Mar-1984
57 0057 1   Change default behavior of SHOW QUEUE/OUTPUT.

```


QUEMANSHO
V04-000

SHOW QUEUE command

C 1
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 2
(1)

```

: 58      0058 1 :
: 59      0059 1 :
: 60      0060 1 :
: 61      0061 1 :
: 62      0062 1 :
: 63      0063 1 :
: 64      0064 1 :
: 65      0065 1 :
: 66      0066 1 :
: 67      0067 1 :
: 68      0068 1 :
: 69      0069 1 :
: 70      0070 1 :
: 71      0071 1 :
: 72      0072 1 :
: 73      0073 1 :
: 74      0074 1 :
: 75      0075 1 :
: 76      0076 1 :
: 77      0077 1 :
: 78      0078 1 :
: 79      0079 1 :
: 80      0080 1 :
: 81      0081 1 :
: 82      0082 1 :
: 83      0083 1 :
: 84      0084 1 :
: 85      0085 1 :
: 86      0086 1 :
: 87      0087 1 : **

V03-009 PCG0002      Peter George      29-Feb-1984
      Add support for /OUTPUT qualifier.
      Write values of /GENERIC out separately.

V03-008 PCG0001      Peter George      17-Feb-1984
      Fix bug in wrap code and always clear buffer after writing it.

V03-007 LMP0140      L. Mark Pilant,    24-Aug-1983
      Add support for alphanumeric UICs.

V03-006 MLJ0118      Martin L. Jack, 23-Aug-1983
      Change names to track $QUIDEF, implement page setup.

V03-005 MLJ0115      Martin L. Jack, 30-Jul-1983
      Changes for job controller baselevel.

V03-004 MLJ0114      Martin L. Jack, 23-Jun-1983
      Rewrite using $GETQUI.

V03-003 MLJ0113      Martin L. Jack, 26-May-1983
      Changes for job controller baselevel.

V03-002 MLJ0112      Martin L. Jack, 29-Apr-1983
      Track queue format changes.

V03-001 MLJ0109      Martin L. Jack, 13-Apr-1983
      Support new job and queue states and track queue format changes.
```



```
89 0088 1 LIBRARY 'SYS$LIBRARY:STARLET';
90 0089 1
91 0090 1
92 0091 1 FORWARD ROUTINE
93 0092 1 SHOWQUEUE: NOVALUE, Main routine
94 0093 1 OPEN_OUTPUT: NOVALUE, Open the output file
95 0094 1 CLOSE_OUTPUT: NOVALUE, Close the output file
96 0095 1 FORMAT_CHARACTERISTICS: NOVALUE, Format characteristics mask
97 0096 1 FORMAT_CPU_TIME: NOVALUE, Format CPU time value
98 0097 1 FORMAT_DEFAULT: NOVALUE, Format /DEFAULT value
99 0098 1 FORMAT_MARGINS: NOVALUE, Format /MARGIN value
100 0099 1 FORMAT_PARAMETERS: NOVALUE, Format /PARAMETERS value
101 0100 1 FORMAT_PROTECTION: NOVALUE, Format /PROTECTION value
102 0101 1 FORMAT_SCHEDULE: NOVALUE, Format /SCHEDULE value
103 0102 1 FORMAT_SEPARATE: NOVALUE, Format /SEPARATE value
104 0103 1 FORMAT_WORKING_SET: NOVALUE, Format working set value
105 0104 1 SHOW_CHARACTERISTICS: NOVALUE, Display characteristics
106 0105 1 SHOW_FORMS: NOVALUE, Display forms
107 0106 1 SHOW_QUEUES: NOVALUE, Display queues
108 0107 1 SHOW_JOBS: NOVALUE, Display jobs
109 0108 1 SHOW_FILES: NOVALUE, Display files
110 0109 1 FAO: NOVALUE, Format information into line
111 0110 1 EOL: NOVALUE, Write line buffer
112 0111 1 CEOL: NOVALUE, EOL if line nonempty
113 0112 1 PUT_OUTPUT: NOVALUE, Write a record to the listing
114 0113 1 EXIT_HANDLER: NOVALUE, Exit handler
115 0114 1
116 0115 1
117 0116 1 EXTERNAL ROUTINE
118 0117 1 CLISGET_VALUE: ADDRESSING MODE(GENERAL),
119 0118 1 ! Get entity value
120 0119 1 CLISPRESENT: ADDRESSING MODE(GENERAL);
121 0120 1 ! Determine if entity is present
122 0121 1
123 0122 1
124 0123 1 LITERAL
125 0124 1 TRUE= 1;
126 0125 1 FALSE= 0;
127 0126 1
128 0127 1
129 0128 1 STRUCTURE
130 0129 1 BBLOCK[O,P,S,E;N]=
131 0130 1 [N]
132 0131 1 (BBLOCK + 0)<P,S,E>;
133 0132 1
134 0133 1
135 0134 1 PSECT
136 0135 1 CODE= CODE,
137 0136 1 PLIT= CODE,
138 0137 1 OWN= DATA,
139 0138 1 GLOBAL= DATA;
140 0139 1
141 0140 1
142 0141 1 LITERAL
143 0142 1 LIST_DEFAULT= 80,
144 0143 1 LIST_SIZE= 132;
145 0144 1
```



```
: 146      0145 1
: 147      0146 1 OWN
: 148      0147 1
: 149      0148 1
: 150      0149 1
: 151      0150 1
: 152      0151 1
: 153      0152 1
: 154      0153 1
: 155      0154 1
: 156      0155 1
: 157      0156 1
: 158      0157 1
: 159      0158 1 MACRO
: 160      0159 1
: 161      0160 1
: 162      0161 1
: 163      0162 1
: 164      0163 1
: 165      0164 1
: 166      0165 1
: 167      0166 1
: 168      0167 1
: 169      0168 1 MACRO
: 170      M 0169 1
: 171      M 0170 1
: 172      0171 1
: 173      0172 1
: 174      M 0173 1
: 175      M 0174 1
: 176      M 0175 1
: 177      M 0176 1
: 178      M 0177 1
: 179      0178 1
: 180      0179 1
: 181      M 0180 1
: 182      0181 1
: 183      0182 1
: 184      M 0183 1
: 185      0184 1
: 186      0185 1
: 187      0186 1
: 188      P 0187 1
: 189      P 0188 1
: 190      P 0189 1
: 191      0190 1
: 192      0191 1
: 193      0192 1 BUILTIN
: 194      0193 1
: 195      0194 1
: 196      0195 1
: 197      0196 1

      FLAGS:
      LIST_DESC:
      LIST_BUFFER:
      P2_DESC:
      FAB:
      NAM:
      RAB:
      OUTPUT_DESC:
      OUTPUT_BUFF:

      BBLOCK[8],
      VECTOR[2],
      VECTOR[LIST_SIZE, BYTE],
      BBLOCK[DSCSR_D_BLN], ! Descriptor for P2
      $FAB_DECL,
      $NAM_DECL,
      $RAB_DECL,
      BBLOCK[DSCSC_S_BLN],
      BBLOCK[NAMSC_MAXRSS];

      V_FILES= 0,0,1,0 %;
      V_FULL= 0,1,1,0 %;
      V_TRUNCATE= 0,2,1,0 %;
      B_LEFT_MARGIN= 4,0,8,0 %;
      B_RIGHT_MARGIN= 5,0,8,0 %;
      B_PENDING_LINES= 6,0,8,0 %;
      B_PENDING_SPACES= 7,0,8,0 %;

      FAO_(A)=
      -FAO(UPLIT BYTE (%ASCIC A)
      %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING) %,

      FAOL_(A)=
      BEGIN
      FAO(UPLIT BYTE (%ASCIC A)
      %IF NOT %NULL(%REMAINING) %THEN , %FI %REMAINING);
      EOL();
      END %,

      EOL_(A)=
      EOL() %,

      CEOL_(A)=
      CEOL() %;

      $SHR_MSGDEF(QUEMAN, 251, GLOBAL,
      (openout, severe),
      (closeout, severe),
      (writeerr, severe));

      EMUL,
      SKPC,
      TESTBITCS,
      TESTBITSC;
```



```
: 199 0197 1 GLOBAL ROUTINE SHOWQUEUE: NOVALUE=
: 200 0198 1
: 201 0199 1 !++
: 202 0200 1
: 203 0201 1 FUNCTIONAL DESCRIPTION:
: 204 0202 1 This routine implements the SHOW QUEUE command.
: 205 0203 1
: 206 0204 1 INPUT PARAMETERS:
: 207 0205 1 NONE
: 208 0206 1
: 209 0207 1 IMPLICIT INPUTS:
: 210 0208 1 Command qualifiers.
: 211 0209 1
: 212 0210 1 OUTPUT PARAMETERS:
: 213 0211 1 NONE
: 214 0212 1
: 215 0213 1 IMPLICIT OUTPUTS:
: 216 0214 1 NONE
: 217 0215 1
: 218 0216 1 ROUTINE VALUE:
: 219 0217 1 NONE
: 220 0218 1
: 221 0219 1 SIDE EFFECTS:
: 222 0220 1 NONE
: 223 0221 1
: 224 0222 1 --
: 225 0223 1
: 226 0224 2 BEGIN
: 227 0225 2 LOCAL
: 228 0226 2 EXIT_BLOCK: VECTOR[5]; ! Exit control block
: 229 0227 2
: 230 0228 2
: 231 0229 2 ! Open the output file and get its width.
: 232 0230 2
: 233 0231 2 OPEN_OUTPUT();
: 234 0232 2
: 235 0233 2
: 236 0234 2 ! Initialize the listing descriptor.
: 237 0235 2
: 238 0236 2 LIST_DESC[0] = .FLAGS[B_RIGHT_MARGIN];
: 239 0237 2 LIST_DESC[1] = LIST_BUFFER;
: 240 0238 2
: 241 0239 2
: 242 0240 2 ! Process the command qualifiers.
: 243 0241 2
: 244 0242 2 FLAGS[V_FILES] = CLISPRESNT($DESCRIPTOR('FILES'));
: 245 0243 2 FLAGS[V_FULL] = CLISPRESNT($DESCRIPTOR('FULL'));
: 246 0244 2
: 247 0245 2
: 248 0246 2 ! Get the P2 parameter.
: 249 0247 2
: 250 0248 2 P2_DESC[DSC$W_LENGTH] = 0;
: 251 0249 2 P2_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 252 0250 2 P2_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
: 253 0251 2 P2_DESC[DSC$A_POINTER] = 0;
: 254 0252 2 CLISGET_VALUE($DESCRIPTOR('P2'), P2_DESC);
: 255 0253 2
```



```

: 256      0254      2      ! Ensure that pending wildcarding is erased.
: 257      0255      2      !
: 258      0256      2      !
: 259      0257      2      $GETQUIW(FUNC=QUI$_CANCEL_OPERATION);
: 260      0258      2
: 261      0259      2
: 262      0260      2      ! Establish an exit handler to cancel the open wildcarding on an abnormal exit.
: 263      0261      2      !
: 264      0262      2      EXIT_BLOCK[0] = 0;
: 265      0263      2      EXIT_BLOCK[1] = EXIT_HANDLER;
: 266      0264      2      EXIT_BLOCK[2] = 1;
: 267      0265      2      EXIT_BLOCK[3] = EXIT_BLOCK[4];
: 268      0266      2      EXIT_BLOCK[4] = 0;
: 269      0267      2      $DCLEXH(DESBLK=EXIT_BLOCK);
: 270      0268      2
: 271      0269      2
: 272      0270      2      ! Display the defined characteristics if requested.
: 273      0271      2      !
: 274      0272      2      IF CL$PRESENT($DESCRIPTOR('CHARACTERISTICS'))
: 275      0273      2      THEN
: 276      0274      2          SHOW_CHARACTERISTICS()
: 277      0275      2
: 278      0276      2
: 279      0277      2      ! Display the defined forms if requested.
: 280      0278      2      !
: 281      0279      2      ELSE IF CL$PRESENT($DESCRIPTOR('FORMS'))
: 282      0280      2      THEN
: 283      0281      2          SHOW_FORMS()
: 284      0282      2
: 285      0283      2
: 286      0284      2      ! Display queues if requested.
: 287      0285      2      !
: 288      0286      2      ELSE
: 289      0287      2          SHOW_QUEUES();
: 290      0288      2
: 291      0289      2
: 292      0290      2      ! Close the output file.
: 293      0291      2      !
: 294      0292      2      CLOSE_OUTPUT();
: 295      0293      2
: 296      0294      2
: 297      0295      2      ! Cancel the exit handler.
: 298      0296      2      !
: 299      0297      2      $CANEXH(DESBLK=EXIT_BLOCK);
: 300      0298      1      END;
```

```

.TITLE QUEMANSHO SHOW QUEUE command
.IDENT \V04-000\
```

```

.PSECT DATA,NOEXE,2
```

```

00000 FLAGS: .BLKB 8
00008 LIST_DESC:
          .BLKB 8
00010 LIST_BUFFER:
          .BLKB 132
```


00094 P2_DESC: .BLKB 8
0009C FAB: .BLKB 80
000EC NAM: .BLKB 96
0014C RAB: .BLKB 68
00190 OUTPUT_DESC: .BLKB 8
00198 OUTPUT_BUFF: .BLKB 255

.PSECT CODE, NOWRT, 2

53 45 4C 49 46 00000 P.AAB: .ASCII \FILES\
00005 .BLKB 3
00000005 00008 P.AAA: .LONG 5
00000000' 0000C .ADDRESS P.AAB
4C 4C 55 46 00010 P.AAD: .ASCII \FULL\
00000004 00014 P.AAC: .LONG 4
00000000' 00018 .ADDRESS P.AAD
32 50 0001C P.AAF: .ASCII \P2\
0001E .BLKB 2
00000002 00020 P.AAE: .LONG 2
00000000' 00024 .ADDRESS P.AAF
53 43 49 54 53 49 52 45 54 43 41 52 41 48 43 00028 P.AAH: .ASCII \CHARACTERISTICS\
00037 .BLKB 1
0000000F 00038 P.AAG: .LONG 15
00000000' 0003C .ADDRESS P.AAH
53 4D 52 4F 46 00040 P.AAJ: .ASCII \FORMS\
00045 .BLKB 3
00000005 00048 P.AAI: .LONG 5
00000000' 0004C .ADDRESS P.AAJ

QUEMAN\$_OPENOUT== 16453796
QUEMAN\$_CLOSEOUT== 16453724
QUEMAN\$_WRITEERR== 16453844
.EXTRN CLISGET VALUE, CLISPRESENT
.EXTRN SYSSGETQUIW, SYSSDCLEXH
.EXTRN SYSSCANEXH

001C 00000 .ENTRY SHOWQUEUE, Save R2,R3,R4
54 B3 AF 9E 00002 MOVAB P.AAA, R4
53 00000000G 00 9E 00006 MOVAB CLISPRESENT, R3
52 0000' CF 9E 0000D MOVAB FLAGS, R2
5E 14 C2 00012 SUBL2 #20, SP
0000V CF 00 FB 00015 CALLS #0, OPEN_OUTPUT
08 A2 05 A2 9A 0001A MOVZBL FLAGS+5, LIST_DESC
0C A2 10 A2 9E 0001F MOVAB LIST_BUFFER, LIST_DESC+4
63 54 DD 00024 PUSHL R4
00 01 FB 00026 CALLS #1, CLISPRESENT
00 50 FO 00029 INSV R0, #0, #1, FLAGS
0C A4 9F 0002E PUSHAB P.AAC
63 01 FB 00031 CALLS #1, CLISPRESENT
01 50 FO 00034 INSV R0, #1, #1, FLAGS
0094 C2 020E0000 8F D0 00039 MOVL #34471936, P2_DESC
0098 C2 D4 00042 CLRL P2_DESC+4
0094 C2 9F 00046 PUSHAB P2_DESC
18 A4 9F 0004A PUSHAB P.AAE
00000000G 00 02 FB 0004D CALLS #2, CLISGET_VALUE

0197
0231
0236
0237
0242
0243
0248
0251
0252

QUEMANSHO
V04-000

SHOW QUEUE command

I 1
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 8
(3)

			7E	7C	00054	CLRQ	-(SP)	: 0257
			7E	7C	00056	CLRQ	-(SP)	
	7E		01	7D	00058	MOVQ	#1, -(SP)	
			7E	D4	0005B	CLRL	-(SP)	
00000000G	00		07	FB	0005D	CALLS	#7, SYSS\$GETQUIW	
			6E	D4	00064	CLRL	EXIT_BLOCK	: 0262
04	AE	0000V	CF	9E	00066	MOVAB	EXIT_HANDLER, EXIT_BLOCK+4	: 0263
08	AE		01	D0	0006C	MOVL	#1, EXIT_BLOCK+8	: 0264
0C	AE	10	AE	9E	00070	MOVAB	EXIT_BLOCK+16, EXIT_BLOCK+12	: 0265
		10	AE	D4	00075	CLRL	EXIT_BLOCK+16	: 0266
			5E	DD	00078	PUSHL	SP	: 0267
00000000G	00		01	FB	0007A	CALLS	#1, SYSS\$DCLEXH	
		30	A4	9F	00081	PUSHAB	P.AAG	: 0272
	63		01	FB	00084	CALLS	#1, CLIS\$PRESENT	
	07		50	E9	00087	BLBC	R0, 1\$	
0000V	CF		00	FB	0008A	CALLS	#0, SHOW_CHARACTERISTICS	: 0274
			15	11	0008F	BRB	3\$	
		40	A4	9F	00091	PUSHAB	P.AAI	: 0279
	63		01	FB	00094	CALLS	#1, CLIS\$PRESENT	
	07		50	E9	00097	BLBC	R0, 2\$	
0000V	CF		00	FB	0009A	CALLS	#0, SHOW_FORMS	: 0281
			05	11	0009F	BRB	3\$	
0000V	CF		00	FB	000A1	CALLS	#0, SHOW_QUEUES	: 0287
0000V	CF		00	FB	000A6	CALLS	#0, CLOSE_OUTPUT	: 0292
			5E	DD	000AB	PUSHL	SP	: 0297
00000000G	00		01	FB	000AD	CALLS	#1, SYSS\$CANEXH	
			04	000B4	RET			: 0298

; Routine Size: 181 bytes, Routine Base: CODE + 0050


```
302 0299 1 ROUTINE OPEN_OUTPUT: NOVALUE=
303 0300 1
304 0301 1 ++
305 0302 1
306 0303 1 FUNCTIONAL DESCRIPTION:
307 0304 1 This routine opens the output file.
308 0305 1
309 0306 1 INPUT PARAMETERS:
310 0307 1 NONE
311 0308 1
312 0309 1 IMPLICIT INPUTS:
313 0310 1 /OUTPUT qualifier.
314 0311 1
315 0312 1 OUTPUT PARAMETERS:
316 0313 1 NONE
317 0314 1
318 0315 1 IMPLICIT OUTPUTS:
319 0316 1 NONE
320 0317 1
321 0318 1 ROUTINE VALUE:
322 0319 1 NONE
323 0320 1
324 0321 1 SIDE EFFECTS:
325 0322 1 NONE
326 0323 1
327 0324 1 --
328 0325 1
329 0326 2 BEGIN
330 0327 2 LOCAL
331 0328 2 DEVBUFSIZ: VOLATILE,
332 0329 2 DEVNAM: BBLOCK[DSC$C_S_BLN],
333 0330 2 ITMLST: BBLOCK[16],
334 0331 2 STATUS;
335 0332 2
336 0333 2
337 0334 2 ! Get the name of the output file.
338 0335 2
339 0336 2 IF CL$PRESENT($DESCRIPTOR('OUTPUT'))
340 0337 2 THEN
341 0338 2 BEGIN
342 0339 2 OUTPUT_DESC[DSC$W_LENGTH] = 0;
343 0340 2 OUTPUT_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
344 0341 2 OUTPUT_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
345 0342 2 OUTPUT_DESC[DSC$A_POINTER] = 0;
346 0343 2 CL$GET_VALUE($DESCRIPTOR('OUTPUT'), OUTPUT_DESC);
347 0344 2 END
348 0345 2 ELSE
349 0346 2 BEGIN
350 0347 2 OUTPUT_DESC[DSC$W_LENGTH] = 11;
351 0348 2 OUTPUT_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
352 0349 2 OUTPUT_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
353 0350 2 OUTPUT_DESC[DSC$A_POINTER] = UPLIT BYTE (ASCII 'SYSS$OUTPUT:');
354 0351 2 END;
355 0352 2
356 0353 2
357 0354 2 ! Initialize the FAB and NAM blocks.
358 0355 2
```



```
359 P 0356 2 $FAB_INIT(FAB=FAB,  
360 P 0357 2     FNS=.OUTPUT_DESC[DSC$W_LENGTH],  
361 P 0358 2     FNA=.OUTPUT_DESC[DSC$A_POINTER],  
362 P 0359 2     FOP=<OFF,SQD>,  
363 P 0360 2     DNM='.LIS',  
364 P 0361 2     RAT=CR,  
365 P 0362 2     FAC=PUT,  
366 P 0363 2     NAM=NAM);  
367 P 0364 2  
368 P 0365 2 $NAM_INIT(NAM=NAM,  
369 P 0366 2     ESA=OUTPUT_BUFF,  
370 P 0367 2     ESS=NAM$C_MAXRSS,  
371 P 0368 2     RSA=OUTPUT_BUFF,  
372 P 0369 2     RSS=NAM$C_MAXRSS);  
373 P 0370 2  
374 P 0371 2  
375 P 0372 2 ! Open the output file and save the file name away.  
376 P 0373 2 !  
377 P 0374 2 STATUS = $CREATE(FAB=FAB);  
378 P 0375 2  
379 P 0376 2 IF .NAM[NAM$B_RSL] NEQ 0  
380 P 0377 2 THEN  
381 P 0378 2 BEGIN  
382 P 0379 2     OUTPUT_DESC[DSC$W_LENGTH] = .NAM[NAM$B_RSL];  
383 P 0380 2     OUTPUT_DESC[DSC$A_POINTER] = OUTPUT_BUFF;  
384 P 0381 2 END  
385 P 0382 2 ELSE IF .NAM[NAM$B_ESL] NEQ 0  
386 P 0383 2 THEN  
387 P 0384 2 BEGIN  
388 P 0385 2     OUTPUT_DESC[DSC$W_LENGTH] = .NAM[NAM$B_ESL];  
389 P 0386 2     OUTPUT_DESC[DSC$A_POINTER] = OUTPUT_BUFF;  
390 P 0387 2 END;  
391 P 0388 2  
392 P 0389 2 IF NOT .STATUS  
393 P 0390 2 THEN  
394 P 0391 2     SIGNAL_STOP(QUEMAN$_OPENOUT,1,OUTPUT_DESC,.STATUS,.FAB[FAB$$_STV]);  
395 P 0392 2  
396 P 0393 2  
397 P 0394 2 ! Initialize the RAB and connect the record stream.  
398 P 0395 2 !  
399 P 0396 2 $RAB_INIT(RAB=RAB,  
400 P 0397 2     RAC=SEQ,  
401 P 0398 2     FAB=FAB);  
402 P 0399 2  
403 P 0400 2 STATUS = $CONNECT(RAB=RAB);  
404 P 0401 2 IF NOT .STATUS  
405 P 0402 2 THEN  
406 P 0403 2     SIGNAL_STOP(QUEMAN$_OPENOUT,1,OUTPUT_DESC,.STATUS,.RAB[RAB$$_STV]);  
407 P 0404 2  
408 P 0405 2  
409 P 0406 2 ! Do a $GETDVI to get the width  
410 P 0407 2 !  
411 P 0408 2 DEVNAM[DSC$W_LENGTH] = .NAM[NAM$B_DEV];  
412 P 0409 2 DEVNAM[DSC$A_POINTER] = .NAM[NAM$C_DEV];  
413 P 0410 2 ITMLST[0,0,16,0] = %ALLOCATION(DEVBUFSIZ);  
414 P 0411 2 ITMLST[2,0,16,0] = DVI$DEVBUFSIZ;  
415 P 0412 2 ITMLST[4,0,32,0] = DEVBUFSIZ;
```



```
: 416      0413  2 ITMLST[8,0,32,0] = 0;  
: 417      0414  2 ITMLST[12,0,32,0] = 0;  
: 418      0415  2 FLAGS[B_RIGHT_MARGIN] = LIST_DEFAULT;  
: 419      0416  3 IF $GETDVIW(DEVNAM=DEVNAM, ITMLST=ITMLST)  
: 420      0417  2 THEN  
: 421      0418  2     FLAGS[B_RIGHT_MARGIN] = MINU(.DEVBUFSIZ, %ALLOCATION(LIST_BUFFER));  
: 422      0419  2  
: 423      0420  1 END;
```

```
54 55 50 54 55 4F 00105 P.AAL: .ASCII \OUTPUT\  
0010B .BLKB 1  
00000006 0010C P.AAK: .LONG 6  
00000000 00110 .ADDRESS P.AAL  
54 55 50 54 55 4F 00114 P.AAN: .ASCII \OUTPUT\  
0011A .BLKB 2  
00000006 0011C P.AAM: .LONG 6  
00000000 00120 .ADDRESS P.AAN  
3A 54 55 50 54 55 4F 24 53 59 53 00124 P.AAO: .ASCII \SYSS$OUTPUT\  
53 49 4C 2E 0012F P.AAP: .ASCII \.LIS\  
:
```

```
$RMS_PTR=  
$RMS_PTR=  
$RMS_PTR=  
FAB  
NAM  
RAB  
.EXTRN SYSS$CREATE, SYSS$CONNECT  
.EXTRN SYSS$GETDVIW
```

01FC 00000 OPEN_OUTPUT:

```
58 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8 : 0299  
57 0000' CF 9E 00009 MOVAB LIB$STOP, R8 :  
5E 1C C2 0000E MOVAB OUTPUT_DESC, R7 :  
C5 AF 9F 00011 SUBL2 #28, SP :  
00000000G 00 01 FB 00014 PUSHAB P.AAK : 0336  
18 50 E9 0001B CALLS #1, CLIS$PRESENT :  
67 020E0000 8F D0 0001E BLBC R0, 1$ : 0339  
04 A7 D4 00025 MOVL #34471936, OUTPUT_DESC : 0342  
57 DD 0002B CLRL OUTPUT_DESC+4 : 0343  
BC AF 9F 0002A PUSHAB P.AAM :  
00000000G 00 02 FB 0002D CALLS #2, CLIS$GET_VALUE : 0336  
OC 11 00034 BRB 2$ : 0347  
67 010E000B 8F D0 00036 1$: MOVL #17694731, OUTPUT_DESC : 0350  
04 A7 9E 0003D MOVAB P.AAO, OUTPUT_DESC+4 : 0363  
6E 00 2C 00042 2$: MOVC5 #0, (SP), #0, #80, $RMS_PTR :  
FF0C C7 5003 8F B0 0004C MOVW #20483, $RMS_PTR :  
FF10 C7 20000040 8F D0 00053 MOVL #536870976, $RMS_PTR+4 :  
FF22 C7 01 90 0005C MOVW #1, $RMS_PTR+22 :  
FF2A C7 0202 8F B0 00061 MOVW #514, $RMS_PTR+30 :  
FF34 C7 FF5C C7 9E 00068 MOVAB NAM, $RMS_PTR+40 :  
FF38 C7 04 A7 D0 0006F MOVL OUTPUT_DESC+4, $RMS_PTR+44 :  
FF3C C7 84 AF 9E 00075 MOVAB P.AAP, $RMS_PTR+48 :  
FF40 C7 67 90 0007B MOVL OUTPUT_DESC, $RMS_PTR+52 :  
FF41 C7 04 90 00080 MOVW #4, $RMS_PTR+53 :  
0060 8F 00 6E 00 2C 00085 MOVC5 #0, (SP), #0, #96, $RMS_PTR : 0369  
FF5C C7 0008C
```


	FF5C	C7	6002	8F	B0	0008F	MOVW	#24578, \$RMS_PTR	
	FF5E	C7		01	8E	00096	MNEGB	#1, \$RMS_PTR+2	
	FF60	C7	08	A7	9E	0009B	MOVAB	OUTPUT_BUFF, \$RMS_PTR+4	
	FF66	C7		01	8E	000A1	MNEGB	#1, \$RMS_PTR+10	
	FF68	C7	08	A7	9E	000A6	MOVAB	OUTPUT_BUFF, \$RMS_PTR+12	
			FF0C	C7	9F	000AC	PUSHAB	FAB	0374
00000000G	00			01	FB	000B0	CALLS	#1, SYSS\$CREATE	
	56			50	D0	000B7	MOVL	R0, STATUS	
	50	FF5F		C7	9A	000BA	MOVZBL	NAM+3, R0	0376
				07	12	000BF	BNEQ	3\$	
	50	FF67		C7	9A	000C1	MOVZBL	NAM+11, R0	0382
				08	13	000C6	BEQL	4\$	
	67			50	B0	000C8	MOVW	R0, OUTPUT_DESC	0385
04	A7	08		A7	9E	000CB	MOVAB	OUTPUT_BUFF, OUTPUT_DESC+4	0386
	13			56	E8	000D0	BLBS	STATUS, 5\$	0389
		FF18		C7	DD	000D3	PUSHL	FAB+12	0391
				56	DD	000D7	PUSHL	STATUS	
				57	DD	000D9	PUSHL	R7	
				01	DD	000DB	PUSHL	#1	
		00FB10A4		8F	DD	000DD	PUSHL	#16453796	
	68			05	FB	000E3	CALLS	#5, LIB\$STOP	
0044	8F	00		00	2C	000E6	MOVCS	#0, (SP), #0, #68, \$RMS_PTR	0398
			BC	A7		000ED			
	BC	A7	4401	8F	B0	000EF	MOVW	#17409, \$RMS_PTR	
			DA	A7	94	000F5	CLRB	\$RMS_PTR+30	
	F8	A7	FF0C	C7	9E	000F8	MOVAB	FAB, \$RMS_PTR+60	
			BC	A7	9F	000FE	PUSHAB	RAB	0400
00000000G	00			01	FB	00101	CALLS	#1, SYSS\$CONNECT	
	56			50	D0	00108	MOVL	R0, STATUS	
	12			56	E8	0010B	BLBS	STATUS, 6\$	0401
		C8		A7	DD	0010E	PUSHL	RAB+12	0403
				56	DD	00111	PUSHL	STATUS	
				57	DD	00113	PUSHL	R7	
				01	DD	00115	PUSHL	#1	
		00FB10A4		8F	DD	00117	PUSHL	#16453796	
	68			05	FB	0011D	CALLS	#5, LIB\$STOP	
	10	AE	95	A7	9B	00120	MOVZBW	NAM+57, DEVNAM	0408
	14	AE	A0	A7	D0	00125	MOVL	NAM+68, DEVNAM+4	0409
		6E	00080004	8F	D0	0012A	MOVL	#524292, ITMLST	0410
	04	AE		18	AE	00131	MOVAB	DEVBUFSIZ, ITMLST+4	0412
				08	AE	00136	CLRQ	ITMLST+8	0413
FE75	C7			50	8F	00139	MOVB	#80, FLAGS+5	0415
					7E	0013F	CLRQ	-(SP)	0416
					7E	00141	CLRQ	-(SP)	
					7E	00143	CLRQ	-(SP)	
			10	AE	9F	00143	PUSHAB	ITMLST	
			24	AE	9F	00146	PUSHAB	DEVNAM	
				7E	7C	00149	CLRQ	-(SP)	
00000000G	00			08	FB	0014B	CALLS	#8, SYSS\$GETDVIW	
	16			50	E9	00152	BLBC	R0, 8\$	
	50		18	AE	D0	00155	MOVL	DEVBUFSIZ, R0	0418
00000084	8F			50	D1	00159	CMPL	R0, #132	
				04	1B	00160	BLEQU	7\$	
	50		84	8F	9A	00162	MOVZBL	#132, R0	
FE75	C7			50	90	00166	MOVB	R0, FLAGS+5	
				04	0016B	8\$:	RET		0420

; Routine Size: 364 bytes, Routine Base: CODE + 0133

QUEMANSHO
V04-000

SHOW QUEUE command

N 1
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 13
(4)


```

: 425      0421 1 ROUTINE CLOSE_OUTPUT: NOVALUE=
: 426      0422 1
: 427      0423 1 ++
: 428      0424 1
: 429      0425 1 FUNCTIONAL DESCRIPTION:
: 430      0426 1 This routine closes the output file.
: 431      0427 1
: 432      0428 1 INPUT PARAMETERS:
: 433      0429 1 NONE
: 434      0430 1
: 435      0431 1 IMPLICIT INPUTS:
: 436      0432 1 Output FAB and RAB.
: 437      0433 1
: 438      0434 1 OUTPUT PARAMETERS:
: 439      0435 1 NONE
: 440      0436 1
: 441      0437 1 IMPLICIT OUTPUTS:
: 442      0438 1 NONE
: 443      0439 1
: 444      0440 1 ROUTINE VALUE:
: 445      0441 1 NONE
: 446      0442 1
: 447      0443 1 SIDE EFFECTS:
: 448      0444 1 NONE
: 449      0445 1
: 450      0446 1 --
: 451      0447 1
: 452      0448 2 BEGIN
: 453      0449 2 LOCAL
: 454      0450 2 STATUS:
: 455      0451 2
: 456      0452 2
: 457      0453 2 ! Disconnect the record stream.
: 458      0454 2
: 459      0455 3 IF NOT (STATUS = $DISCONNECT(RAB=RAB))
: 460      0456 2 THEN
: 461      0457 2 SIGNAL(QUEMAN$_CLOSEOUT,1,OUTPUT_DESC,,STATUS,,RAB[RAB$_L_STV]);
: 462      0458 2
: 463      0459 2
: 464      0460 2 ! Close the file.
: 465      0461 2
: 466      0462 3 IF NOT (STATUS = $CLOSE(FAB=FAB))
: 467      0463 2 THEN
: 468      0464 2 SIGNAL(QUEMAN$_CLOSEOUT,1,OUTPUT_DESC,,STATUS,,FAB[FAB$_L_STV]);
: 469      0465 2
: 470      0466 1 END;

```

.EXTRN SYSS\$DISCONNECT, SYSS\$CLOSE

```

                                001C 00000 CLOSE_OUTPUT:
                                .WORD Save R2,R3,R4
                                MOVAB LIB$SIGNAL, R4
                                MOVAB OUTPUT_DESC, R3
                                PUSHAB RAB
                                CALLS #1, SYSS$DISCONNECT

```

```

: 0421
:
: 0455
:

```


QUEMANSHO
V04-000

SHOW QUEUE command

C 2
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 15
(5)

52		50	DO 00018	MOVL	R0, STATUS	
12		52	E8 0001B	BLBS	STATUS, 1\$	
	C8	A3	DD 0001E	PUSHL	RAB+12	0457
		52	DD 00021	PUSHL	STATUS	
		53	DD 00023	PUSHL	R3	
		01	DD 00025	PUSHL	#1	
	00FB105C	8F	DD 00027	PUSHL	#16453724	
64		05	FB 0002D	CALLS	#5, LIB\$SIGNAL	
	FF0C	C3	9F 00030 1\$:	PUSHAB	FAB	0462
00000000G	00	01	FB 00034	CALLS	#1, SYSS\$CLOSE	
	52	50	DO 0003B	MOVL	R0, STATUS	
	13	52	E8 0003E	BLBS	STATUS, 2\$	
		C3	DD 00041	PUSHL	FAB+12	0464
	FF18	52	DD 00045	PUSHL	STATUS	
		53	DD 00047	PUSHL	R3	
		01	DD 00049	PUSHL	#1	
	00FB105C	8F	DD 0004B	PUSHL	#16453724	
64		05	FB 00051	CALLS	#5, LIB\$SIGNAL	
		04	00054 2\$:	RET		0466

; Routine Size: 85 bytes, Routine Base: CODE + 029F


```
472 0467 1 ROUTINE FORMAT_CHARACTERISTICS(C): NOVALUE=
473 0468 1
474 0469 1 !++
475 0470 1
476 0471 1 FUNCTIONAL DESCRIPTION:
477 0472 1 This routine formats a characteristics mask.
478 0473 1
479 0474 1 INPUT PARAMETERS:
480 0475 1 C - Pointer to 16-byte characteristics mask.
481 0476 1
482 0477 1 IMPLICIT INPUTS:
483 0478 1 NONE
484 0479 1
485 0480 1 OUTPUT PARAMETERS:
486 0481 1 NONE
487 0482 1
488 0483 1 IMPLICIT OUTPUTS:
489 0484 1 NONE
490 0485 1
491 0486 1 ROUTINE VALUE:
492 0487 1 NONE
493 0488 1
494 0489 1 SIDE EFFECTS:
495 0490 1 NONE
496 0491 1
497 0492 1 --
498 0493 1
499 0494 2 BEGIN
500 0495 2 MAP
501 0496 2 C: REF BITVECTOR; ! Pointer to characteristics vector
502 0497 2 LOCAL
503 0498 2 UNDONE, ! Count of set bits left
504 0499 2 LPUNC, ! Leading punctuation
505 0500 2 TPUNC; ! Trailing punctuation
506 0501 2
507 0502 2
508 0503 2 ! Count the number of characteristics to display.
509 0504 2
510 0505 2 UNDONE = 0;
511 0506 2 INCR I FROM 0 TO 127 DO IF .C[I] THEN UNDONE = .UNDONE + 1;
512 0507 2
513 0508 2
514 0509 2 IF .UNDONE NEQ 0
515 0510 2 THEN
516 0511 3 BEGIN
517 0512 3 LPUNC = UPLIT BYTE (%ASCIC '/CHAR=(');
518 0513 3 INCR I FROM 0 TO 127 DO
519 0514 4 BEGIN
520 0515 4 IF .C[I]
521 0516 4 THEN
522 0517 5 BEGIN
523 0518 5 TPUNC = %ASCIC ',';
524 0519 5 UNDONE = .UNDONE - 1;
525 0520 5 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
526 0521 5 FAO ('!AC!UL!AC', .LPUNC, .I, TPUNC);
527 0522 5 LPUNC = UPLIT BYTE (%ASCIC '(');
528 0523 4 END;
```


QUEMANSHO
V04-000

SHOW QUEUE command

E 2
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 17
(6)

```
: 529      0524 4      IF .UNDONE EQL 0 THEN EXITLOOP;
: 530      0525 3      END;
: 531      0526 3      FLAGS[B_PENDING_SPACES] = 1;
: 532      0527 2      END;
: 533      0528 1      END;
```

```
43 41 28 3D 52 41 48 43 2F 07 002F4 P.AAQ: .ASCII <7>\ /CHAR=(\
21 4C 55 21 43 41 21 09 002FC P.AAR: .ASCII <9>\ !AC!UL!AC\
00 00306 P.AAS: .ASCII <0>
```

				001C 00000	FORMAT_CHARACTERISTICS:	
		5E	04 C2 00002	.WORD	Save R2,R3,R4	0467
			54 D4 00005	SUBL2	#4, SP	0505
			50 D4 00007	CLRL	UNDONE	0506
02	04	BC	50 E1 00009 1\$:	CLRL	I, ac, 2\$	
			54 D6 0000E	INCL	UNDONE	
F1		50 0000007F	8F F3 00010 2\$:	AOBLEQ	#127, I, 1\$	
			54 D5 00018	TSTL	UNDONE	0509
			3C 13 0001A	BEQL	7\$	
		53 CE	AF 9E 0001C	MOVAB	P.AAQ, LPUNC	0512
			52 D4 00020	CLRL	I	0513
20	04	BC	52 E1 00022 3\$:	BBC	I, ac, 5\$	0515
		6E 2C01	8F 3C 00027	MOVZWL	#11265, TPUNC	0518
			54 D7 0002C	DECL	UNDONE	0519
			05 12 0002E	BNEQ	4\$	0520
		6E 2901	8F 3C 00030	MOVZWL	#10497, TPUNC	
		4004	8F BB 00035 4\$:	PUSHR	#^M<R2,SP>	0521
			53 DD 00039	PUSHL	LPUNC	
		B7	AF 9F 0003B	PUSHAB	P.AAR	
	0000V	CF	04 FB 0003E	CALLS	#4, FAO	
		53 B9	AF 9E 00043	MOVAB	P.AAS, LPUNC	0522
			54 D5 00047 5\$:	TSTL	UNDONE	0524
			08 13 00049	BEQL	6\$	
CF		52 0000007F	8F F3 0004B	AOBLEQ	#127, I, 3\$	0513
	0000'	CF	01 90 00053 6\$:	MOVB	#1, FLAGS+7	0526
			04 00058 7\$:	RET		0528

; Routine Size: 89 bytes, Routine Base: CODE + 0307


```
535 0529 1 ROUTINE FORMAT_CPU_TIME(NAME,VALUE): NOVALUE=
536 0530 1
537 0531 1 ++
538 0532 1
539 0533 1 FUNCTIONAL DESCRIPTION:
540 0534 1 This routine formats a CPU time parameter.
541 0535 1
542 0536 1 INPUT PARAMETERS:
543 0537 1 NAME - Name of the CPU time parameter.
544 0538 1 VALUE - Value of the parameter.
545 0539 1
546 0540 1 IMPLICIT INPUTS:
547 0541 1 NONE
548 0542 1
549 0543 1 OUTPUT PARAMETERS:
550 0544 1 NONE
551 0545 1
552 0546 1 IMPLICIT OUTPUTS:
553 0547 1 NONE
554 0548 1
555 0549 1 ROUTINE VALUE:
556 0550 1 NONE
557 0551 1
558 0552 1 SIDE EFFECTS:
559 0553 1 NONE
560 0554 1
561 0555 1 --
562 0556 1
563 0557 2 BEGIN
564 0558 2 IF .VALUE EQL 0
565 0559 2 THEN
566 0560 2 FAO_('!/AC=INFINITE', .NAME)
567 0561 2 ELSE
568 0562 3 BEGIN
569 0563 3 LOCAL
570 0564 3 TIME_STRING: BBLOCK[16],
571 0565 3 TIME_DESC: VECTOR[2],
572 0566 3 DELTA: VECTOR[2];
573 0567 3
574 0568 3 ! Convert the time expressed in 10 ms units to a system delta-time value.
575 0569 3 !
576 0570 3 EMUL(%REF(-100000), VALUE, %REF(0), DELTA);
577 0571 3
578 0572 3 ! Convert the system delta-time value to a string.
579 0573 3 !
580 0574 3 TIME_DESC[0] = %ALLOCATION(TIME_STRING);
581 0575 3 TIME_DESC[1] = TIME_STRING;
582 0576 3 SASCTIM(TIMBUF=TIME_DESC, TIMADR=DELTA);
583 0577 3
584 0578 3 ! Edit the string to make it more readable. If the string specifies zero
585 0579 3 ! days or zero hundredths of seconds, remove these fields from the string
586 0580 3 ! since they are rarely nonzero. Remove leading spaces from the string.
587 0581 3 ! Also, convert the space that separates days from hours to a hyphen to
588 0582 3 ! match the delta-time input syntax.
589 0583 3
590 0584 3
591 0585 3
```



```
: 592      0586      3      1
: 593      0587      3      1
: 594      0588      3      1
: 595      0589      3      1
: 596      0590      3      1
: 597      0591      3      1
: 598      0592      2      1
: 599      0593      2      1
: 600      0594      1      1

      IF .TIME_STRING[0,0,32,0] EQL ' 0' THEN TIME_STRING[3,0,8,0] = %C' ';
      IF .TIME_STRING[13,0,24,0] EQL '.00' THEN TIME_DESC[0] = 13;
      SKPC(%REF(%C' '), TIME_DESC[0], .TIME_DESC[1]; TIME_DESC[0], TIME_DESC[1]);
      TIME_STRING[4,0,8,0] = %C' -';
      FAO T' /!AC=!AS', .NAME, TIME_DESC);
      END;
      FLAGS[B_PENDING_SPACES] = 1;
      END;
```

```
45 54 49 4E 49 46 4E 49 3D 43 41 21 2F 0D 00360 P.AAT: .ASCII <13>\!/AC=INFINITE\
53 41 21 3D 43 41 21 2F 08 0036E P.AAU: .ASCII <8>\!/AC=!AS\
```

.EXTRN SYSSASCTIM

0000 00000 FORMAT_CPU TIME:

5E	20	C2	00002	.WORD	Save nothing	0529
	08	AC	D5	SUBL2	#32, SP	0558
		0D	12	TSTL	VALUE	
	04	AC	DD	BNEQ	1\$	0560
	D9	AF	9F	PUSHL	NAME	
		02	FB	PUSHAB	P.AAT	
0000V	CF	0E	FB	CALLS	#2, FAO	
		5E	11	BRB	4\$	
6E	00	08	AC	EMUL	#-100000, VALUE, #0, DELTA	0571
		08	AE	MOVL	#16, TIME_DESC	0576
		0C	AE	MOVAB	TIME_STRING, TIME_DESC+4	0577
			7E	CLRL	-(SP)	0578
		04	AE	PUSHAB	DELTA	
		10	AE	PUSHAB	TIME_DESC	
		7E	D4	CLRL	-(SP)	
	00000000G	00	04	CALLS	#4, SYSSASCTIM	
	30202020	8F	10	CMPL	TIME_STRING, #807411744	0587
			04	BNEQ	2\$	
0030302E	8F	1D	AE	MOVB	#32, TIME_STRING+3	
			00	CMPZV	#0, #24, TIME_STRING+13, #3158062	0588
			04	BNEQ	3\$	
		08	AE	MOVL	#13, TIME_DESC	
		0C	BE	SKPC	#32, TIME_DESC, @TIME_DESC+4	0589
		08	AE	MOVQ	R0, TIME_DESC	
		14	AE	MOVB	#45, TIME_STRING+4	0590
			08	PUSHAB	TIME_DESC	0591
		04	AC	PUSHL	NAME	
		87	AF	PUSHAB	P.AAU	
	0000V	CF	03	CALLS	#3, FAO	
	0000'	CF	01	MOVB	#1, FLAGS+7	0593
			04	RET		0594

; Routine Size: 123 bytes, Routine Base: CODE + 0377


```
602 0595 1 ROUTINE FORMAT_DEFAULT(QUEUE_FLAGS): NOVALUE=
603 0596 1
604 0597 1 !++
605 0598 1
606 0599 1 FUNCTIONAL DESCRIPTION:
607 0600 1 This routine formats the value of the /DEFAULT qualifier.
608 0601 1
609 0602 1 INPUT PARAMETERS:
610 0603 1 QUEUE_FLAGS - Queue flags item.
611 0604 1
612 0605 1 IMPLICIT INPUTS:
613 0606 1 NONE
614 0607 1
615 0608 1 OUTPUT PARAMETERS:
616 0609 1 NONE
617 0610 1
618 0611 1 IMPLICIT OUTPUTS:
619 0612 1 NONE
620 0613 1
621 0614 1 ROUTINE VALUE:
622 0615 1 NONE
623 0616 1
624 0617 1 SIDE EFFECTS:
625 0618 1 NONE
626 0619 1
627 0620 1 --
628 0621 1
629 0622 2 BEGIN
630 0623 2 MAP
631 0624 2 QUEUE_FLAGS: BBLOCK; ! Queue flags
632 0625 2 LOCAL
633 0626 2 UNDONE: BBLOCK[4], ! What's left to do
634 0627 2 LPUNC;
635 0628 2 TPUNC;
636 0629 2 MACRO
637 0630 2 V_BURST= 0,0,1,0 %;
638 0631 2 V_FEED= 0,1,1,0 %;
639 0632 2 V_FLAG= 0,2,1,0 %;
640 0633 2 V_TRAILER= 0,3,1,0 %;
641 0634 2
642 0635 2
643 0636 2 UNDONE = 0;
644 0637 2 IF .QUEUE_FLAGS[QUI$V-QUEUE-FILE-BURST]
645 0638 2 OR .QUEUE_FLAGS[QUI$V-QUEUE-FILE-BURST_ONE]
646 0639 2 THEN UNDONE[V_BURST] = TRUE;
647 0640 2 IF .QUEUE_FLAGS[QUI$V-QUEUE-FILE-PAGINATE]
648 0641 2 THEN UNDONE[V_FEED] = TRUE;
649 0642 2 IF .QUEUE_FLAGS[QUI$V-QUEUE-FILE-FLAG]
650 0643 2 OR .QUEUE_FLAGS[QUI$V-QUEUE-FILE-FLAG_ONE]
651 0644 2 THEN UNDONE[V_FLAG] = TRUE;
652 0645 2 IF .QUEUE_FLAGS[QUI$V-QUEUE-FILE-TRAILER]
653 0646 2 OR .QUEUE_FLAGS[QUI$V-QUEUE-FILE-TRAILER_ONE]
654 0647 2 THEN UNDONE[V_TRAILER] = TRUE;
655 0648 2
656 0649 2
657 0650 2 IF .UNDONE NEQ 0
658 0651 2 THEN
```



```
: 659      0652 3 BEGIN
: 660      0653 3 LPUNC = UPLIT BYTE (%ASCIC '/DEFAULT=(');
: 661      0654 3 WHILE .UNDONE NEQ 0 DO
: 662      0655 4 BEGIN
: 663      0656 4 TPUNC = %ASCIC ',';
: 664      0657 4
: 665      0658 4
: 666      0659 4 IF TESTBITSC(UNDONE[V_BURST])
: 667      0660 4 THEN
: 668      0661 5 BEGIN
: 669      0662 5 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 670      0663 5 IF .QUEUE_FLAGS[QUI$V QUEUE_FILE_BURST ONE]
: 671      0664 5 THEN FAO_('!ACBURST=ONE!AC', .LPUNC, TPUNC)
: 672      0665 5 ELSE FAO_('!ACBURST!AC', .LPUNC, TPUNC);
: 673      0666 5 END
: 674      0667 5
: 675      0668 5
: 676      0669 4 ELSE IF TESTBITSC(UNDONE[V_FEED])
: 677      0670 4 THEN
: 678      0671 5 BEGIN
: 679      0672 5 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 680      0673 5 IF .QUEUE_FLAGS[QUI$V QUEUE_FILE_PAGINATE]
: 681      0674 5 THEN FAO_('!ACFEED!AC', .LPUNC, TPUNC);
: 682      0675 5 END
: 683      0676 5
: 684      0677 5
: 685      0678 4 ELSE IF TESTBITSC(UNDONE[V_FLAG])
: 686      0679 4 THEN
: 687      0680 5 BEGIN
: 688      0681 5 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 689      0682 5 IF .QUEUE_FLAGS[QUI$V QUEUE_FILE_FLAG ONE]
: 690      0683 5 THEN FAO_('!ACFLAG=ONE!AC', .LPUNC, TPUNC)
: 691      0684 5 ELSE FAO_('!ACFLAG!AC', .LPUNC, TPUNC);
: 692      0685 5 END
: 693      0686 5
: 694      0687 5
: 695      0688 4 ELSE IF TESTBITSC(UNDONE[V_TRAILER])
: 696      0689 4 THEN
: 697      0690 5 BEGIN
: 698      0691 5 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 699      0692 5 IF .QUEUE_FLAGS[QUI$V QUEUE_FILE_TRAILER ONE]
: 700      0693 5 THEN FAO_('!ACTRAILER=ONE!AC', .LPUNC, TPUNC)
: 701      0694 5 ELSE FAO_('!ACTRAILER!AC', .LPUNC, TPUNC);
: 702      0695 5 END;
: 703      0696 4
: 704      0697 4 LPUNC = UPLIT BYTE (%ASCIC '');
: 705      0698 4 END;
: 706      0699 3 FLAGS[B_PENDING_SPACES] = 1;
: 707      0700 3 END;
: 708      0701 2
: 709      0702 1 END;
```

```
41 21 45 4E 28 3D 54 4C 55 41 46 45 44 2F 0A 003F2 P.AAV: .ASCII <10>\ /DEFAULT=(\
4F 3D 54 53 52 55 42 43 41 21 0F 003FD P.AAW: .ASCII <15>\ !ACBURST=ONE!AC\
43 0040C
```

...

QUEMANSHO
V04-000

SHOW QUEUE command

J 2
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 22
(8)

			43	41	21	54	53	52	55	42	43	41	21	0B	0040D	P.AAX:	.ASCII	<11>\!ACBURST!AC\		
			43	41	21	44	45	45	46	43	41	21	0A	00419	P.AAY:	.ASCII	<10>\!ACFEED!AC\			
43	41	21	45	4E	4F	3D	47	41	4C	46	43	41	21	0E	00424	P.AAZ:	.ASCII	<14>\!ACFLAG=ONE!AC\		
			43	41	21	47	41	4C	46	43	41	21	0A	00433	P.ABA:	.ASCII	<10>\!ACFLAG!AC\			
45	4E	4F	3D	52	45	4C	49	41	52	54	43	41	21	11	0043E	P.ABB:	.ASCII	<17>\!ACTRAILER=ONE!AC\		
			43	41	21	52	45	4C	49	41	52	54	43	41	21	0D	00450	P.ABC:	.ASCII	<13>\!ACTRAILER!AC\
														00	0045E	P.ABD:	.ASCII	<0>		

001C 00000 FORMAT_DEFAULT:

			54	8E	AF	9E	00002	.WORD	Save R2,R3,R4		0595
			5E		04	C2	00006	MOVAB	P.AAV, R4		
					53	D4	00009	SUBL2	#4, SP		
05	04	AC			03	E0	0000B	CLRL	UNDONE		0636
03	04	AC			04	E1	00010	BBS	#3, QUEUE_FLAGS, 1\$		0637
		53			01	88	00015	BBC	#4, QUEUE_FLAGS, 2\$		0638
03	06	AC			06	E1	00018	BISB2	#1, UNDONE		0639
		53			02	88	0001D	BBC	#6, QUEUE_FLAGS+2, 3\$		0640
05	04	AC			05	E0	00020	BISB2	#2, UNDONE		0641
03	04	AC			06	E1	00025	BBS	#5, QUEUE_FLAGS, 4\$		0642
		53			04	88	0002A	BBC	#6, QUEUE_FLAGS, 5\$		0643
				04	AC	95	0002D	BISB2	#4, UNDONE		0644
				03	AC	19	00030	TSTB	QUEUE_FLAGS		0645
		53		05	AC	E9	00032	BLSS	6\$		
					08	88	00036	BLBC	QUEUE_FLAGS+1, 7\$		0646
					53	D5	00039	BISB2	#8, UNDONE		0647
					01	12	0003B	TSTL	UNDONE		0650
						04	0003D	BNEQ	8\$		
		52			64	9E	0003E	RET			
					53	D5	00041	MOVAB	P.AAV, LPUNC		0653
					03	12	00043	TSTL	UNDONE		0654
					0095	31	00045	BNEQ	10\$		
		6E	2C01		8F	3C	00048	BRW	23\$		
20		53			00	E5	0004D	MOVZWL	#11265, TPUNC		0656
					53	D5	00051	BBCC	#0, UNDONE, 13\$		0659
					05	12	00053	TSTL	UNDONE		0662
		6E	2901		8F	3C	00055	BNEQ	11\$		
09	04	AC			04	E1	0005A	MOVZWL	#10497, TPUNC		
					8F	BB	0005F	BBC	#4, QUEUE_FLAGS, 12\$		0663
			4004		A4	9F	00063	PUSHR	#*M<R2,SP\$		0664
			0B		69	11	00066	PUSHAB	P.AAW		
					8F	BB	00068	BRB	21\$		
			4004		A4	9F	0006C	PUSHR	#*M<R2,SP>		0665
			1B		60	11	0006F	PUSHAB	P.AAX		
					01	E5	00071	BRB	21\$		
17		53			53	D5	00075	BBCC	#1, UNDONE, 15\$		0669
					05	12	00077	TSTL	UNDONE		0672
		6E	2901		8F	3C	00079	BNEQ	14\$		
53	06	AC			06	E1	0007E	MOVZWL	#10497, TPUNC		
					8F	BB	00083	BBC	#6, QUEUE_FLAGS+2, 22\$		0673
			4004		A4	9F	00087	PUSHR	#*M<R2,SP\$		0674
			27		45	11	0008A	PUSHAB	P.AAY		
					02	E5	0008C	BRB	21\$		
20		53			53	D5	00090	BBCC	#2, UNDONE, 18\$		0678
								TSTL	UNDONE		0681

QUEMANSHO
V04-000

SHOW QUEUE command

K 2
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 23
(8)

09	04	6E AC	2901	05 8F 06 8F A4 2A	12 3C E1 BB 9F 11	00092 00094 00099 0009E 000A2 000A5	16\$:	BNEQ 16\$ MOVZWL #10497, TPUNC BBC #6, QUEUE FLAGS, 17\$ PUSHR #^M<R2,SP> PUSHAB P.AAZ BRB 21\$	0682 0683
			4004 32	8F A4 2A	BB 9F 11	0009E 000A2 000A5			
			4004 41	8F A4 21	BB 9F 11	000A7 000AB 000AE	17\$:	PUSHR #^M<R2,SP> PUSHAB P.ABA BRB 21\$	0684
22		53		03 53 05	E5 D5 12	000B0 000B4 000B6	18\$:	BBCC #3, UNDONE, 22\$ TSTL UNDONE BNEQ 19\$	0688 0691
		6E 09	2901 05	8F AC	3C E9	000B8 000BD	19\$:	MOVZWL #10497, TPUNC BLBC QUEUE FLAGS+1, 20\$ PUSHR #^M<R2,SP> PUSHAB P.ABB	0692 0693
			4004 4C	8F A4 07	BB 9F 11	000C1 000C5 000C8		BRB 21\$ PUSHR #^M<R2,SP> PUSHAB P.ABC	
			4004 5E	8F A4 03	BB 9F FB	000CA 000CE 000D1	20\$:	CALLS #3, FAO MOVAB P.ABD, LPUNC BRW 9\$	0694
	0000V	CF	6C	FF A4 01	64 9E 90	000DA 000DD 000E2	21\$: 22\$: 23\$:	MOVAB P.ABD, LPUNC BRW 9\$ MOVB #1, FLAGS+7 RET	0698 0654 0700 0702

; Routine Size: 227 bytes, Routine Base: CODE + 045F


```
711 0703 1 ROUTINE FORMAT_MARGINS(BOTTOM,LEFT,RIGHT,TOP): NOVALUE=
712 0704 1
713 0705 1 ++
714 0706 1
715 0707 1 FUNCTIONAL DESCRIPTION:
716 0708 1 This routine formats the value of the /MARGIN qualifier.
717 0709 1
718 0710 1 INPUT PARAMETERS:
719 0711 1 BOTTOM - Values of the four margins
720 0712 1 LEFT -
721 0713 1 RIGHT -
722 0714 1 TOP -
723 0715 1
724 0716 1 IMPLICIT INPUTS:
725 0717 1 NONE
726 0718 1
727 0719 1 OUTPUT PARAMETERS:
728 0720 1 NONE
729 0721 1
730 0722 1 IMPLICIT OUTPUTS:
731 0723 1 NONE
732 0724 1
733 0725 1 ROUTINE VALUE:
734 0726 1 NONE
735 0727 1
736 0728 1 SIDE EFFECTS:
737 0729 1 NONE
738 0730 1
739 0731 1 --
740 0732 1
741 0733 2 BEGIN
742 0734 2 LOCAL
743 0735 2 UNDONE: BBLOCK[4], ! What's left to do
744 0736 2 LPUNC;
745 0737 2 TPUNC;
746 0738 2 MACRO
747 0739 2 V_TOP= 0.0,1.0 %;
748 0740 2 V_BOTTOM= 0.1,1.0 %;
749 0741 2 V_LEFT= 0.2,1.0 %;
750 0742 2 V_RIGHT= 0.3,1.0 %;
751 0743 2
752 0744 2
753 0745 2 UNDONE = 0;
754 0746 2 IF .TOP NEQ 0
755 0747 2 THEN UNDONE[V_TOP] = TRUE;
756 0748 2 IF .BOTTOM NEQ 0
757 0749 2 THEN UNDONE[V_BOTTOM] = TRUE;
758 0750 2 IF .LEFT NEQ 0
759 0751 2 THEN UNDONE[V_LEFT] = TRUE;
760 0752 2 IF .RIGHT NEQ 0
761 0753 2 THEN UNDONE[V_RIGHT] = TRUE;
762 0754 2
763 0755 2
764 0756 2 IF .UNDONE NEQ 0
765 0757 2 THEN
766 0758 3 BEGIN
767 0759 3 LPUNC = UPLIT BYTE (%ASCIC '/MARGIN=(');
```

```
: 768      0760 3      WHILE .UNDONE NEQ 0 DO
: 769      0761 4          BEGIN
: 770      0762 4              TPUNC = %ASCIC ',,';
: 771      0763 4
: 772      0764 4
: 773      0765 4          IF TESTBITSC(UNDONE[V_TOP])
: 774      0766 4              THEN
: 775      0767 5              BEGIN
: 776      0768 5                  IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 777      0769 5                  FAO_('!ACTOP=!UL!AC', .LPUNC, .TOP, TPUNC);
: 778      0770 5                  END
: 779      0771 5
: 780      0772 5
: 781      0773 4          ELSE IF TESTBITSC(UNDONE[V_BOTTOM])
: 782      0774 4              THEN
: 783      0775 5              BEGIN
: 784      0776 5                  IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 785      0777 5                  FAO_('!ACBOTTOM=!UL!AC', .LPUNC, .BOTTOM, TPUNC);
: 786      0778 5                  END
: 787      0779 5
: 788      0780 5
: 789      0781 4          ELSE IF TESTBITSC(UNDONE[V_LEFT])
: 790      0782 4              THEN
: 791      0783 5              BEGIN
: 792      0784 5                  IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 793      0785 5                  FAO_('!ACLEFT=!UL!AC', .LPUNC, .LEFT, TPUNC);
: 794      0786 5                  END
: 795      0787 5
: 796      0788 5
: 797      0789 4          ELSE IF TESTBITSC(UNDONE[V_RIGHT])
: 798      0790 4              THEN
: 799      0791 5              BEGIN
: 800      0792 5                  IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 801      0793 5                  FAO_('!ACRIGHT=!UL!AC', .LPUNC, .RIGHT, TPUNC);
: 802      0794 4                  END;
: 803      0795 4
: 804      0796 4
: 805      0797 4              LPUNC = UPLIT BYTE (%ASCIC ');
: 806      0798 3              END;
: 807      0799 3          FLAGS[B_PENDING_SPACES] = 1;
: 808      0800 2          END;
: 809      0801 1      END;
```

```
21 43 41 21 4C 55 21 3D 4D 4F 54 54 4F 42 43 41 21 10 0055A P.ABG: .ASCII <16>\!ACBOTTOM=!UL!AC\
43 41 21 4C 55 21 3D 54 46 45 4C 43 41 21 0E 0056B P.ABH: .ASCII <14>\!ACLEFT=!UL!AC\
41 21 4C 55 21 3D 54 48 47 49 52 43 41 21 0F 0057A P.ABI: .ASCII <15>\!ACRIGHT=!UL!AC\
                                43 00589
                                00 0058A P.ABJ: .ASCII <0>
```

001C 00000 FORMAT_MARGINS:

[illegible]

QUEMANSHO
V04-000

SHOW QUEUE command

B 3
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 27
(9)

0000V	CF		04	FB	0009D	13\$:	CALLS	#4, FAO
	53	48	A4	9E	000A2	14\$:	MOVAB	P.ABJ, LPUNC
			8A	11	000A6		BRB	5\$
0000'	CF		01	90	000A8	15\$:	MOVB	#1, FLAGS+7
			04	000AD	16\$:		RET	

: 0797
: 0760
: 0799
: 0801

; Routine Size: 174 bytes, Routine Base: CODE + 058B


```
811 0802 1 ROUTINE FORMAT_PARAMETERS: NOVALUE=
812 0803 1
813 0804 1 ++
814 0805 1
815 0806 1 FUNCTIONAL DESCRIPTION:
816 0807 1 This routine formats the value of the /PARAMETERS qualifier.
817 0808 1
818 0809 1 INPUT PARAMETERS:
819 0810 1 Pairs of <length, address>.
820 0811 1
821 0812 1 IMPLICIT INPUTS:
822 0813 1 NONE
823 0814 1
824 0815 1 OUTPUT PARAMETERS:
825 0816 1 NONE
826 0817 1
827 0818 1 IMPLICIT OUTPUTS:
828 0819 1 NONE
829 0820 1
830 0821 1 ROUTINE VALUE:
831 0822 1 NONE
832 0823 1
833 0824 1 SIDE EFFECTS:
834 0825 1 NONE
835 0826 1
836 0827 1 --
837 0828 1
838 0829 2 BEGIN
839 0830 2 BUILTIN
840 0831 2 ACTUALPARAMETER;
841 0832 2 LOCAL
842 0833 2 UNDONE,
843 0834 2 LPUNC,
844 0835 2 TPUNC;
845 0836 2
846 0837 2
847 0838 2 UNDONE = 8;
848 0839 2 DECR I FROM 15 TO 1 BY 2 DO
849 0840 3 BEGIN
850 0841 3 IF ACTUALPARAMETER(.I) NEQ 0 THEN EXITLOOP;
851 0842 3 UNDONE = .UNDONE - 1;
852 0843 3 END;
853 0844 2
854 0845 2
855 0846 2 IF .UNDONE NEQ 0
856 0847 2 THEN
857 0848 3 BEGIN
858 0849 3 LPUNC = UPLIT BYTE (%ASCIC '/PARAM=(');
859 0850 3 INCR I FROM 1 TO 15 BY 2 DO
860 0851 4 BEGIN
861 0852 4 TPUNC = %ASCIC ',';
862 0853 4 UNDONE = .UNDONE - 1;
863 0854 4 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
864 0855 4 FAO_(!AC"!AF"!AC',
865 0856 4 .LPUNC,
866 0857 4 ACTUALPARAMETER(.I), ACTUALPARAMETER(.I + 1),
867 0858 4 TPUNC);
```

```
: 868      0859 4      LPUNC = UPLIT BYTE (%ASCII '');  
: 869      0860 4      IF .UNDONE EQL 0 THEN EXITLOOP;  
: 870      0861 3      END;  
: 871      0862 3      FLAGS[B_PENDING_SPACES] = 1;  
: 872      0863 2      END;  
: 873      0864 1 END;
```

```
43 41 21 28 3D 4D 41 52 41 50 2F 08 00639 P.ABK: .ASCII <8>\PARAM=(\  
22 46 41 21 22 43 41 21 0B 00642 P.ABL: .ASCII <11>\!AC''!AF''!AC\  
00 0064E P.ABM: .ASCII <0>
```

003C 00000 FORMAT_PARAMETERS:

	5E	04	C2	00002	.WORD	Save R2,R3,R4,R5	0802
	53	08	D0	00005	SUBL2	#4, SP	
	50	0F	D0	00008	MOVL	#8, UNDONE	0838
		6C40	D5	0000B	MOVL	#15, I	0839
		09	12	0000E	TSTL	(AP)[I]	0841
		53	D7	00010	BNEQ	2\$	
FFF2	50	FE	8F	01	DECL	UNDONE	0842
				01	ACBB	#1, #-2, I, 1\$	0839
				53	TSTL	UNDONE	0846
				40	BEQL	6\$	
	55	CA	AF	9E	MOVAB	P.ABK, LPUNC	0849
	52		01	D0	MOVL	#1, I	0850
	6E	2C01	8F	3C	MOVZWL	#11265, TPUNC	0852
			53	D7	DECL	UNDONE	0853
			54	D4	CLRL	R4	0854
			53	D5	TSTL	UNDONE	
			07	12	BNEQ	4\$	
			54	D6	INCL	R4	
	6E	2901	8F	3C	MOVZWL	#10497, TPUNC	
			5E	DD	PUSHL	SP	0858
		04	AC42	DD	PUSHL	4(AP)[I]	
			6C42	DD	PUSHL	(AP)[I]	
			55	DD	PUSHL	LPUNC	
		AD	AF	9F	PUSHAB	P.ABL	
	0000V	CF	05	FB	CALLS	#5, FAO	
			B1	AF	MOVAB	P.ABM, LPUNC	0859
			54	E8	BLBS	R4, 5\$	0860
FFCC	52		0F	F1	ACBL	#15, #2, I, 3\$	0850
	0000'	CF	01	90	MOVB	#1, FLAGS+7	0862
			04	0C05D	RET		0864

; Routine Size: 94 bytes, Routine Base: CODE + 064F


```

875 0865 1 ROUTINE FORMAT_PROTECTION(VALUE): NOVALUE=
876 0866 1
877 0867 1 |++
878 0868 1
879 0869 1 | FUNCTIONAL DESCRIPTION:
880 0870 1 |     This routine formats a protection value.
881 0871 1 |
882 0872 1 | INPUT PARAMETERS:
883 0873 1 |     VALUE          - Value of the parameter.
884 0874 1 |
885 0875 1 | IMPLICIT INPUTS:
886 0876 1 |     NONE
887 0877 1 |
888 0878 1 | OUTPUT PARAMETERS:
889 0879 1 |     NONE
890 0880 1 |
891 0881 1 | IMPLICIT OUTPUTS:
892 0882 1 |     NONE
893 0883 1 |
894 0884 1 | ROUTINE VALUE:
895 0885 1 |     NONE
896 0886 1 |
897 0887 1 | SIDE EFFECTS:
898 0888 1 |     NONE
899 0889 1 |
900 0890 1 | --
901 0891 1
902 0892 2 BEGIN
903 0893 2 LOCAL
904 0894 2     LPUNC,
905 0895 2     TPUNC;
906 0896 2 BIND
907 0897 2     SOGW = UPLIT('SOGW'): VECTOR[.BYTE],
908 0898 2     RWED = UPLIT('RWED'): VECTOR[.BYTE];
909 0899 2
910 0900 2
911 0901 2 LPUNC = UPLIT BYTE (%ASCIC '/PROTECTION=(');
912 0902 2 INCR I FROM 0 TO 3 DO
913 0903 3 BEGIN
914 0904 3     TPUNC = %ASCIC ',';
915 0905 3     IF .I EQL 3 THEN TPUNC = %ASCIC ')';
916 0906 3     IF .VALUE<.I*4, 4> NEQ %B'1111'
917 0907 3     THEN
918 0908 4 BEGIN
919 0909 4 LOCAL
920 0910 4     T:          VECTOR[4,BYTE],
921 0911 4     TL;
922 0912 4
923 0913 4     TL = 0;
924 0914 4     INCR J FROM 0 TO 3 DO
925 0915 5 BEGIN
926 0916 5     IF NOT .VALUE<.I*4+.J, 1>
927 0917 5     THEN
928 0918 6 BEGIN
929 0919 6     T[.TL] = .RWED[.J];
930 0920 6     TL = .TL + 1;
931 0921 5 END;
```

```

: 932      0922  4      END;
: 933      0923  4      FAO_('!AC!AD:!AD!AC', .LPUNC, 1, SOGW[.I], .TL, T, TPUNC);
: 934      0924  4      END
: 935      0925  3      ELSE
: 936      0926  3      FAO_('!AC!AD!AC', .LPUNC, 1, SOGW[.I], TPUNC);
: 937      0927  3
: 938      0928  3      LPUNC = UPLIT BYTE (%ASCIC '');
: 939      0929  2      END;
: 940      0930  2      FLAGS[B_PENDING_SPACES] = 1;
: 941      0931  1      END;
```

28	3D	4E	4F	49	54	43	45	54	4F	57	47	4F	53	006AD	P.ABN:	.BLKB	3	
43	41	21	44	41	21	3A	44	41	21	44	45	57	52	006B0	P.ABO:	.ASCII	\SOGW\	
				43	41	21	44	41	21	43	41	21	0D	006B4	P.ABP:	.ASCII	\RWED\	
													0D	006B8	P.ABQ:	.ASCII	<13>\PROTECTION=(\	
													09	006C6	P.ABR:	.ASCII	<13>\!AC!AD:!AD!AC\	
													00	006D4	P.ABS:	.ASCII	<9>\!AC!AD!AC\	
														00	006DE	P.ABS:	.ASCII	<0>

SOGW=
RWED=P.ABN
P.ABO

00FC 00000 FORMAT_PROTECTION:

			57	D4	AF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7		0865
			5E		08	C2	00006	MOVAB	P.ABP, R7		
			55		67	9E	00009	SUBL2	#8, SP		0901
					52	D4	0000C	MOVAB	P.ABP, LPUNC		0902
		04	AE	2C01	8F	3C	0000E	CLRL	I		0904
			03		52	D1	00014	MOVZWL	#11265, TPUNC		0905
					06	12	00017	CMPL	I, #3		
		04	AE	2901	8F	3C	00019	BNEQ	2\$		
			56		02	78	00024	MOVZWL	#10497, TPUNC		0923
50		04	54		02	78	00024	MOVAB	SOGW[.I], R6		0906
			52		54	EF	00028	ASHL	#2, I, R4		
			04		50	D1	0002E	EXTZV	R4, #4, VALUE, R0		
			0F		2D	13	00031	CMPL	R0, #15		
					50	7C	00033	BEQL	5\$		
					50	C1	00035	CLRQ	J		0914
			53		50	C1	00035	ADDL3	J, R4, R3		0916
		04	06		53	E0	00039	BBS	R3, VALUE, 4\$		
					03	F3	00044	MOVAB	RWED[J], I[SP]		0919
			ED		04	AE	9F	00048	AOBLEQ	#3, J, 3\$	0914
					04	AE	9F	0004B	PUSHAB	TPUNC	0923
						51	DD	0004E	PUSHAB	T	
						56	DD	00050	PUSHL	TL	
						01	DD	00052	PUSHL	R6	
						55	DD	00054	PUSHL	#1	
						07	9F	00056	PUSHL	LPUNC	
		0000V	CF		0E	A7	FB	00059	PUSHAB	P.ABQ	
						11	11	0005E	CALLS	#7, FAO	
					04	AE	9F	00060	BRB	6\$	0906
						56	DD	00063	PUSHAB	TPUNC	0926
						01	DD	00065	PUSHL	R6	
									PUSHL	#1	

16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 32
(11)

95	0000V	CF	1C	55	DD	00067		PUSHL	LPUNC
				A7	9F	00069		PUSHAB	P.ABR
				05	FB	0006C		CALLS	#5, FAO
			26	A7	9E	00071	6\$:	MOVAB	P.ABS, LPUNC
				03	F3	00075		AOBLEQ	#3, I, 1\$
	0000'	CF		01	90	00079		MOVB	#1, FLAGS+7
					04	0007E		RET	

; Routine Size: 127 bytes, Routine Base: CODE + 06DF

```
0932 1 ROUTINE FORMAT_SCHEDULE(QUEUE_FLAGS): NOVALUE=
0933 1
0934 1 !++
0935 1
0936 1 FUNCTIONAL DESCRIPTION:
0937 1 This routine formats the value of the /SCHEDULE qualifier.
0938 1
0939 1 INPUT PARAMETERS:
0940 1 QUEUE_FLAGS - Queue flags item
0941 1
0942 1 IMPLICIT INPUTS:
0943 1 NONE
0944 1
0945 1 OUTPUT PARAMETERS:
0946 1 NONE
0947 1
0948 1 IMPLICIT OUTPUTS:
0949 1 NONE
0950 1
0951 1 ROUTINE VALUE:
0952 1 NONE
0953 1
0954 1 SIDE EFFECTS:
0955 1 NONE
0956 1
0957 1 --
0958 1
0959 2 BEGIN
0960 2 MAP
0961 2 QUEUE_FLAGS: BBLOCK; ! Queue flags item
0962 2 LOCAL
0963 2 UNDONE: BBLOCK[4], ! What's left to do
0964 2 LPUNC,
0965 2 TPUNC;
0966 2 MACRO
0967 2 V_SIZE= 0,0,1,0 %;
0968 2
0969 2
0970 2 UNDONE = 0;
0971 2 IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_JOB_SIZE_SCHED]
0972 2 THEN UNDONE[V_SIZE] = TRUE;
0973 2
0974 2
0975 2 IF .UNDONE NEQ 0
0976 2 THEN
0977 2 BEGIN
0978 2 LPUNC = UPLIT BYTE (%ASCIC '/SCHEDULE=(');
0979 2 WHILE .UNDONE NEQ 0 DO
0980 2 BEGIN
0981 2 TPUNC = %ASCIC ',';
0982 2
0983 2
0984 2 IF TESTBITSC(UNDONE[V_SIZE])
0985 2 THEN
0986 2 BEGIN
0987 2 IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
0988 2 FAO_('!ACNOSIZE!AC', .LPUNC, TPUNC);
```


QUEMANSHO
V04-000

SHOW QUEUE command

1 3
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 34
(12)

```
: 1000      0989  4      END;
: 1001      0990  4
: 1002      0991  4
: 1003      0992  4      LPUNC = UPLIT BYTE (%ASCIC '');
: 1004      0993  3      END;
: 1005      0994  3      FLAGS[B_PENDING_SPACES] = 1;
: 1006      0995  2      END;
: 1007      0996  1 END;
```

```
43 28 3D 45 4C 55 44 45 48 43 53 2F 0B 0075E P.ABT: .ASCII <11>\ /SCHEDULE=(\
41 41 21 45 5A 49 53 4F 4E 43 41 21 0C 0076A P.ABU: .ASCII <12>\ !ACNOSIZE!AC\
00 00777 P.ABV: .ASCII <0>
```

000C 00000 FORMAT_SCHEDULE:

	5E		04 C2 00002	.WORD	Save R2,R3	0932
			53 D4 00005	SUBL2	#4, SP	
03	05	AC	05 E0 00007	CLRL	UNDONE	0970
		53	01 88 0000C	BBS	#5, QUEUE FLAGS+1, 1\$	0971
			53 D5 0000F 1\$:	BISB2	#1, UNDONE	0972
			31 13 00011	TSTL	UNDONE	0975
	52		AF 9E 00013	BEQL	6\$	
		D0	53 D5 00017 2\$:	MOVAB	P.ABT, LPUNC	0978
			24 13 00019	TSTL	UNDONE	0979
			8F 3C 0001B	BEQL	5\$	
15	6E	2C01	00 E5 00020	MOVZWL	#11265, TPUNC	0981
	53		53 D5 00024	BBCC	#0, UNDONE, 4\$	0984
			05 12 00026	TSTL	UNDONE	0987
	6E	2901	8F 3C 00028	BNEQ	3\$	
		4004	8F BB 0002D 3\$:	MOVZWL	#10497, TPUNC	
		BE	AF 9F 00031	PUSHR	#^M<R2,SP>	0988
	0000V	CF	03 FB 00034	PUSHAB	P.ABU	
		52	AF 9E 00039 4\$:	CALLS	#3, FA0	
			D8 11 0003D	MOVAB	P.ABV, LPUNC	0992
	0000'	CF	01 90 0003F 5\$:	BRB	2\$	0979
			04 00044 6\$:	MOVB	#1, FLAGS+7	0994
				RET		0996

; Routine Size: 69 bytes, Routine Base: CODE + 0778

```
1009 0997 1 ROUTINE FORMAT_SEPARATE(QUEUE_FLAGS,JOB_RESET_MODULES_LENGTH,JOB_RESET_MODULES): NOVALUE=
1010 0998 1
1011 0999 1 ++
1012 1000 1
1013 1001 1 FUNCTIONAL DESCRIPTION:
1014 1002 1 This routine formats the value of the /SEPARATE qualifier.
1015 1003 1
1016 1004 1 INPUT PARAMETERS:
1017 1005 1 QUEUE_FLAGS - Queue flags item
1018 1006 1 JOB_RESET_MODULES_LENGTH - Descriptor for job reset modules item
1019 1007 1 JOB_RESET_MODULES -
1020 1008 1
1021 1009 1 IMPLICIT INPUTS:
1022 1010 1 NONE
1023 1011 1
1024 1012 1 OUTPUT PARAMETERS:
1025 1013 1 NONE
1026 1014 1
1027 1015 1 IMPLICIT OUTPUTS:
1028 1016 1 NONE
1029 1017 1
1030 1018 1 ROUTINE VALUE:
1031 1019 1 NONE
1032 1020 1
1033 1021 1 SIDE EFFECTS:
1034 1022 1 NONE
1035 1023 1
1036 1024 1 --
1037 1025 1
1038 1026 2 BEGIN
1039 1027 2 MAP
1040 1028 2 QUEUE_FLAGS: BBLOCK; ! Queue flags item
1041 1029 2 LOCAL
1042 1030 2 UNDONE: BBLOCK[4], ! What's left to do
1043 1031 2 LPUNC,
1044 1032 2 TPUNC;
1045 1033 2 MACRO
1046 1034 2 V_BURST= 0,0,1,0 %;
1047 1035 2 V_FLAG= 0,1,1,0 %;
1048 1036 2 V_RESET= 0,2,1,0 %;
1049 1037 2 V_TRAILER= 0,3,1,0 %;
1050 1038 2
1051 1039 2
1052 1040 2 UNDONE = 0;
1053 1041 2 IF .QUEUE_FLAGS[QUI$V QUEUE JOB BURST]
1054 1042 2 THEN UNDONE[V BURST] = TRUE;
1055 1043 2 IF .QUEUE_FLAGS[QOI$V QUEUE JOB FLAG]
1056 1044 2 THEN UNDONE[V FLAG] = TRUE;
1057 1045 2 IF .JOB_RESET_MODULES_LENGTH NEQ 0
1058 1046 2 THEN UNDONE[V RESET] = TRUE;
1059 1047 2 IF .QUEUE_FLAGS[QOI$V QUEUE JOB TRAILER]
1060 1048 2 THEN UNDONE[V_TRAILER] = TRUE;
1061 1049 2
1062 1050 2
1063 1051 2 IF .UNDONE NEQ 0
1064 1052 2 THEN
1065 1053 2 BEGIN
```



```
: 1066      1054  3    LPUNC = UPLIT BYTE (%ASCIC '/SEPARATE=());
: 1067      1055  3    WHILE .UNDONE NEQ 0 DO
: 1068      1056  4      BEGIN
: 1069      1057  4        TPUNC = %ASCIC ',';
: 1070      1058  4
: 1071      1059  4
: 1072      1060  4      IF TESTBITSC(UNDONE[V_BURST])
: 1073      1061  4      THEN
: 1074      1062  5        BEGIN
: 1075      1063  5          IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 1076      1064  5          FAO_('!ACBURST!AC', .LPUNC, TPUNC);
: 1077      1065  5          END
: 1078      1066  5
: 1079      1067  5
: 1080      1068  4      ELSE IF TESTBITSC(UNDONE[V_FLAG])
: 1081      1069  4      THEN
: 1082      1070  5        BEGIN
: 1083      1071  5          IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 1084      1072  5          FAO_('!ACFLAG!AC', .LPUNC, TPUNC);
: 1085      1073  5          END
: 1086      1074  5
: 1087      1075  5
: 1088      1076  4      ELSE IF TESTBITSC(UNDONE[V_RESET])
: 1089      1077  4      THEN
: 1090      1078  5        BEGIN
: 1091      1079  5          IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 1092      1080  5          FAO_('!ACRESET=(!AF)!AC',
: 1093      1081  5            .LPUNC,
: 1094      1082  5            .JOB_RESET_MODULES_LENGTH, .JOB_RESET_MODULES,
: 1095      1083  5            TPUNC);
: 1096      1084  5          END
: 1097      1085  5
: 1098      1086  5
: 1099      1087  4      ELSE IF TESTBITSC(UNDONE[V_TRAILER])
: 1100      1088  4      THEN
: 1101      1089  5        BEGIN
: 1102      1090  5          IF .UNDONE EQL 0 THEN TPUNC = %ASCIC ')';
: 1103      1091  5          FAO_('!ACTRAILER!AC', .LPUNC, TPUNC);
: 1104      1092  4          END;
: 1105      1093  4
: 1106      1094  4
: 1107      1095  4      LPUNC = UPLIT BYTE (%ASCIC '');
: 1108      1096  3      END;
: 1109      1097  3      FLAGS[B_PENDING_SPACES] = 1;
: 1110      1098  2      END;
: 1111      1099  1    END;
```

			28	3D	45	54	41	52	41	50	45	53	2F	0B	007BD	P.ABW:	.ASCII	<11>\!SEPARATE=(\		
			43	41	21	54	53	52	55	42	43	41	21	0B	007C9	P.ABX:	.ASCII	<11>\!ACBURST!AC\		
				43	41	21	47	41	4C	46	43	41	21	0A	007D5	P.ABY:	.ASCII	<10>\!ACFLAG!AC\		
29	46	41	21	28	3D	54	45	53	45	52	43	41	21	11	007E0	P.ABZ:	.ASCII	<17>\!ACRESET=(!AF)!AC\		
												43	41	21	007EF					
			43	41	21	52	45	4C	49	41	52	54	43	41	21	0D	007F2	P.ACA:	.ASCII	<13>\!ACTRAILER!AC\
														00	00800	P.ACB:	.ASCII	<0>		

```
001C 00000 FORMAT_SEPARATE:
      54      B7 AF 9E 00002      .WORD      Save R2,R3,R4      : 0997
      5E      04 C2 00006      MOVAB      P.ABW, R4
      52      52 D4 00009      SUBL2      #4, SP
      03      03 E1 0000B      CLRL       UNDONE
      52      01 88 00010      BBC        #3, QUEUE_FLAGS+1, 1$
      03      05 AC 00013 1$:   BISB2      #1, UNDONE
      52      04 E1 00018      BBC        #4, QUEUE_FLAGS+1, 2$
      08      02 88 0001B 2$:   BISB2      #2, UNDONE
      AC      D5 0001E      TSTL       JOB_RESET_MODULES_LENGTH
      03      13 00020      BEQL        3$
      52      04 88 00023 3$:   BISB2      #4, UNDONE
      AC      06 E1 00028 3$:   BBC        #6, QUEUE_FLAGS+1, 4$
      52      08 88 0002B 4$:   BISB2      #8, UNDONE
      52      D5 0002D      TSTL       UNDONE
      7B      13 0002F      BEQL        16$
      53      64 9E 00032 5$:   MOVAB      P.ABW, LPUNC
      52      D5 00034      TSTL       UNDONE
      6E      8F 3C 00036      BEQL        15$
      52      00 E5 0003B      MOVZWL     #11265, TPUNC
      12      52 D5 0003F      BBCC       #0, UNDONE, 7$
      6E      05 12 00041      TSTL       UNDONE
      2901    8F 3C 00043      BNEQ        6$
      4008    8F BB 00048 6$:   MOVZWL     #10497, TPUNC
      0C      A4 9F 0004C      PUSHHR    #^M<R3,SP>
      49      11 0004F      PUSHAB     P.ABX
      12      01 E5 00051 7$:   BRB        13$
      52      D5 00055      BBCC       #1, UNDONE, 9$
      05      12 00057      TSTL       UNDONE
      6E      8F 3C 00059      BNEQ        8$
      4008    8F BB 0005E 8$:   MOVZWL     #10497, TPUNC
      18      A4 9F 00062      PUSHHR    #^M<R3,SP>
      33      11 00065      PUSHAB     P.ABY
      1B      02 E5 00067 9$:   BRB        13$
      52      D5 0006B      BBCC       #2, UNDONE, 11$
      05      12 0006D      TSTL       UNDONE
      6E      8F 3C 0006F      BNEQ        10$
      2901    5E DD 00074 10$:   MOVZWL     #10497, TPUNC
      7E      08 AC 7D 00076      PUSHL      SP
      53      DD 0007A      MOVQ       JOB_RESET_MODULES_LENGTH, -(SP)
      23      A4 9F 0007C      PUSHL      LPUNC
      0000V    CF 05 FB 0007F      PUSHAB     P.ABZ
      15      19 11 00084      CALLS      #5, FAO
      52      03 E5 00086 11$:   BRB        14$
      52      D5 0008A      BBCC       #3, UNDONE, 14$
      05      12 0008C      TSTL       UNDONE
      6E      8F 3C 0008E      BNEQ        12$
      4008    8F BB 00093 12$:   MOVZWL     #10497, TPUNC
      35      A4 9F 00097      PUSHHR    #^M<R3,SP>
      0000V    CF 03 FB 0009A 13$:   PUSHAB     P.ACA
      53      43 A4 9E 0009F 14$:   CALLS      #3, FAO
      0000'    CF 8D 11 000A3      MOVAB      P.ACB, LPUNC
      01      90 000A5 15$:   BRB        5$
      04      000AA 16$:   MOVB       #1, FLAGS+7
      RET
```


QUEMANSHO
V04-000

SHOW QUEUE command

M 3
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 38
(13)

; Routine Size: 171 bytes, Routine Base: CODE + 0801

```

: 1113 1100 1 ROUTINE FORMAT_WORKING_SET(NAME,VALUE): NOVALUE=
: 1114 1101 1
: 1115 1102 1 ++
: 1116 1103 1
: 1117 1104 1 FUNCTIONAL DESCRIPTION:
: 1118 1105 1 This routine formats a working set parameter.
: 1119 1106 1
: 1120 1107 1 INPUT PARAMETERS:
: 1121 1108 1 NAME - Name of the working set parameter.
: 1122 1109 1 VALUE - Value of the parameter.
: 1123 1110 1
: 1124 1111 1 IMPLICIT INPUTS:
: 1125 1112 1 NONE
: 1126 1113 1
: 1127 1114 1 OUTPUT PARAMETERS:
: 1128 1115 1 NONE
: 1129 1116 1
: 1130 1117 1 IMPLICIT OUTPUTS:
: 1131 1118 1 NONE
: 1132 1119 1
: 1133 1120 1 ROUTINE VALUE:
: 1134 1121 1 NONE
: 1135 1122 1
: 1136 1123 1 SIDE EFFECTS:
: 1137 1124 1 NONE
: 1138 1125 1
: 1139 1126 1 --
: 1140 1127 1
: 1141 1128 2 BEGIN
: 1142 1129 2 IF .VALUE EQL 0
: 1143 1130 2 THEN FAO ('/!AC=NONE', .NAME)
: 1144 1131 2 ELSE FAO ('/!AC=!UL', .NAME, .VALUE);
: 1145 1132 2 FLAGS[B_PENDING_SPACES] = 1;
: 1146 1133 1 END;

```

```

45 4E 4F 4E 3D 43 41 21 2F 09 008AC P.ACC: .ASCII <9>\!/AC=NONE\
4C 55 21 3D 43 41 21 2F 08 008B6 P.ACD: .ASCII <8>\!/AC=!UL\

```

```

0000 00000 FORMAT_WORKING_SET:
                                .WORD Save nothing
                                08 AC D5 00002 TSTL VALUE
                                0D 12 00005 BNEQ 1$
                                04 AC DD 00007 PUSHL NAME
                                E0 AF 9F 0000A PUSHAB P.ACC
                                02 FB 0000D CALLS #2, FAO
                                0C 11 00012 BRB 2$
                                04 AC 7D 00014 1$: MOVQ NAME, -(SP)
                                DC AF 9F 00018 PUSHAB P.ACD
                                03 FB 0001B CALLS #3, FAO
                                01 90 00020 2$: MOVB #1, FLAGS+7
                                04 00025 RET

```

; Routine Size: 38 bytes, Routine Base: CODE + 08BF

QUEMANSHO
V04-000

SHOW QUEUE command

B 4
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 40
(14)

```

: 1148      1134 1 ROUTINE SHOW_CHARACTERISTICS: NOVALUE=
: 1149      1135 1
: 1150      1136 1 ++
: 1151      1137 1
: 1152      1138 1 FUNCTIONAL DESCRIPTION:
: 1153      1139 1     This routine displays the defined characteristics.
: 1154      1140 1
: 1155      1141 1 INPUT PARAMETERS:
: 1156      1142 1     NONE
: 1157      1143 1
: 1158      1144 1 IMPLICIT INPUTS:
: 1159      1145 1     NONE
: 1160      1146 1
: 1161      1147 1 OUTPUT PARAMETERS:
: 1162      1148 1     NONE
: 1163      1149 1
: 1164      1150 1 IMPLICIT OUTPUTS:
: 1165      1151 1     NONE
: 1166      1152 1
: 1167      1153 1 ROUTINE VALUE:
: 1168      1154 1     NONE
: 1169      1155 1
: 1170      1156 1 SIDE EFFECTS:
: 1171      1157 1     NONE
: 1172      1158 1
: 1173      1159 1 --
: 1174      1160 1
: 1175      1161 2 BEGIN
: 1176      1162 2 LOCAL
: 1177      1163 2     ITMLST:
: 1178      1164 2     CHARACTERISTIC_NAME:      BBLOCK[52],
: 1179      1165 2     CHARACTERISTIC_NAME_LENGTH:  VOLATILE VECTOR[31,BYTE],
: 1180      1166 2     CHARACTERISTIC_NUMBER:      VOLATILE WORD,
: 1181      1167 2     STATUS,
: 1182      1168 2     IOSB:
: 1183      1169 2     CHARACTERISTIC_PRINTED;      VECTOR[2],
: 1184      1170 2
: 1185      1171 2
: 1186      1172 2 FLAGS[V TRUNCATE] = TRUE;
: 1187      1173 2 CHARACTERISTIC_PRINTED = FALSE;
: 1188      1174 2
: 1189      1175 2
: 1190      1176 2 ! Initialize the item list.
: 1191      1177 2
: 1192      1178 2 ITMLST[0,0,16,0] = .P2_DESC[DESC$W_LENGTH];
: 1193      1179 2 ITMLST[2,0,16,0] = QUI$_SEARCH_NAME;
: 1194      1180 2 ITMLST[4,0,32,0] = .P2_DESC[DESC$A_POINTER];
: 1195      1181 2 ITMLST[8,0,32,0] = 0;
: 1196      1182 2
: 1197      1183 2 ITMLST[12,0,16,0] = 4;
: 1198      1184 2 ITMLST[14,0,16,0] = QUI$_SEARCH_FLAGS;
: 1199      1185 2 ITMLST[16,0,32,0] = UPLIT(QUI$M_SEARCH_WILDCARD);
: 1200      1186 2 ITMLST[20,0,32,0] = 0;
: 1201      1187 2
: 1202      1188 2 ITMLST[24,0,16,0] = %ALLOCATION(CHARACTERISTIC_NAME);
: 1203      1189 2 ITMLST[26,0,16,0] = QUI$ CHARACTERISTIC_NAME;
: 1204      1190 2 ITMLST[28,0,32,0] = CHARACTERISTIC_NAME;

```


63	69	74	73	69	72	65	74	63	61	72	61	68	43	2B	008E5		.BLKB	3	
20	20	20	20	20	20	20	20	20	20	65	6D	61	6E	20	008E8	P.ACE:	.LONG	2	
					75	4E	20	20	20	20	20	20	20	20	008EC	P.ACF:	.ASCII		\+Characteristic name
											72	65	62	6D	008FB				
2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2B	0090A				
															00914		.ASCII	\mber\	
															00918	P.ACG:	.ASCII	\+-----	

20 20 20 20 20 20 20 20 20 20 2D 2D 2D 2D 2D 00927
2D 2D 20 20 20 20 2D 2D 00936
55 36 21 20 20 20 20 20 20 46 41 31 33 21 0F 00940
4C 00944 P.ACH: .ASCII \----\
00953 4C 00953 <15>\!31AF !6UL\

003C 00000 SHOW_CHARACTERISTICS:

55 0000V CF 9E 00002 .WORD Save R2,R3,R4,R5 1134
54 0000V CF 9E 00007 MOVAB EOL, R5
5E 9C AE 9E 0000C MOVAB FAO, R4
0000' CF 04 88 00010 MOVAB -100(SP), SP
30 AE 0000' 53 D4 00015 BISB2 #4, FLAGS 1172
32 AE 4D 8F 9B 00017 CLRL CHARACTERISTIC_PRINTED 1173
34 AE 0000' 53 D4 00015 MOVW P2_DESC, ITMLST 1178
38 AE D4 00028 MOVZBW #77, ITMLST+2 1179
3C AE 004C0004 8F D0 00022 MOVL P2_DESC+4, ITMLST+4 1180
40 AE FF5D CF 9E 00028 CLRL ITMLST+8 1181
44 AE D4 00039 MOVL #4980740, ITMLST+12 1183
48 AE 0005001F 8F D0 00033 MOVAB P.ACE, ITMLST+16 1185
4C AE 10 AE 9E 00039 CLRL ITMLST+20 1186
50 AE 0E AE 9E 0003C MOVL #327711, ITMLST+24 1188
54 AE 00060004 8F D0 00044 MOVAB CHARACTERISTIC_NAME, ITMLST+28 1190
58 AE 08 AE 9E 00049 MOVAB CHARACTERISTIC_NAME_LENGTH, ITMLST+32 1191
5C AE 7C 00056 MOVL #393220, ITMLST+36 1193
7E 7C 00058 MOVAB CHARACTERISTIC_NUMBER, ITMLST+40 1195
08 AE 9F 0005B CLRL ITMLST+44 1196
3C AE 9F 00060 CLRL -(SP) 1208
7E 02 7D 00063 PUSHAB IOSB
00000000G 00 07 FB 00066 PUSHAB ITMLST
52 50 D0 00069 MOVQ #2, -(SP)
03 52 E9 00072 CLRL -(SP)
52 6E D0 00075 CALLS #7, SYSS\$GETQUIW
50 52 D2 00078 MOVL R0, STATUS
24 53 00 E2 0007E BLBC STATUS, 2\$ 1209
0B 50 E9 00082 MOVL IOSB, STATUS
00000000G 00 52 DD 00085 MCOML STATUS, R0 1215
01 FB 00087 BBSS #0, CHARACTERISTIC_PRINTED, 4\$ 1212
2F 11 0008E BLBC R0, 3\$ 1215
FF04 CF 9F 00090 BRB STATUS 1218
64 01 FB 00094 CALLS #1, LIB\$SIGNAL 1217
65 00 FB 00097 BRB 6\$ 1221
FF26 CF 9F 0009A PUSHAB P.ACF
64 01 FB 0009E CALLS #1, FAO
65 00 FB 000A1 CALLS #0, EOL
03 11 000A4 PUSHAB P.ACG 1222
16 50 E8 000A6 CALLS #1, FAO
08 AE DD 000A9 BRB 5\$ 1212
14 AE 9F 000AC BLBS R0, 6\$ 1226
7E 16 AE 3C 000AF PUSHL CHARACTERISTIC_NUMBER 1232
FF39 CF 9F 000B3 PUSHAB CHARACTERISTIC_NAME
64 04 FB 000B7 MOVZWL CHARACTERISTIC_NAME_LENGTH, -(SP)
65 00 FB 000BA CALLS P.ACH
CALLS #4, FAO
CALLS #0, EOL

QUEMANSHO
V04-000

SHOW QUEUE command

F 4
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 44
(15)

0000' CF

9F	11	000BD		BRB	1\$
04	8A	000BF	6\$:	BICB2	#4, FLAGS
04	000C4			RET	

: 1203
: 1236
: 1237

; Routine Size: 197 bytes, Routine Base: CODE + 0954

```
: 1253      1238 1 ROUTINE SHOW_FORMS: NOVALUE=
: 1254      1239 1
: 1255      1240 1 ++
: 1256      1241 1
: 1257      1242 1 FUNCTIONAL DESCRIPTION:
: 1258      1243 1     This routine displays the defined forms.
: 1259      1244 1
: 1260      1245 1 INPUT PARAMETERS:
: 1261      1246 1     NONE
: 1262      1247 1
: 1263      1248 1 IMPLICIT INPUTS:
: 1264      1249 1     NONE
: 1265      1250 1
: 1266      1251 1 OUTPUT PARAMETERS:
: 1267      1252 1     NONE
: 1268      1253 1
: 1269      1254 1 IMPLICIT OUTPUTS:
: 1270      1255 1     NONE
: 1271      1256 1
: 1272      1257 1 ROUTINE VALUE:
: 1273      1258 1     NONE
: 1274      1259 1
: 1275      1260 1 SIDE EFFECTS:
: 1276      1261 1     NONE
: 1277      1262 1
: 1278      1263 1 --
: 1279      1264 1
: 1280      1265 2 BEGIN
: 1281      1266 2 LOCAL
: 1282      1267 2     ITMLST:
: 1283      1268 2     FORM_DESCRIPTION:
: 1284      1269 2     FORM_DESCRIPTION_LENGTH:
: 1285      1270 2     FORM_FLAGS:
: 1286      1271 2     FORM_LENGTH:
: 1287      1272 2     FORM_MARGIN_BOTTOM:
: 1288      1273 2     FORM_MARGIN_LEFT:
: 1289      1274 2     FORM_MARGIN_RIGHT:
: 1290      1275 2     FORM_MARGIN_TOP:
: 1291      1276 2     FORM_NAME:
: 1292      1277 2     FORM_NAME_LENGTH:
: 1293      1278 2     FORM_NUMBER:
: 1294      1279 2     FORM_SETUP_MODULES:
: 1295      1280 2     FORM_SETUP_MODULES_LENGTH:
: 1296      1281 2     FORM_STOCK:
: 1297      1282 2     FORM_STOCK_LENGTH:
: 1298      1283 2     FORM_WIDTH:
: 1299      1284 2     PAGE_SETUP_MODULES:
: 1300      1285 2     PAGE_SETUP_MODULES_LENGTH:
: 1301      1286 2     STATUS,
: 1302      1287 2     IOSB:
: 1303      1288 2     FORM_PRINTED;
: 1304      1289 2
: 1305      1290 2
: 1306      1291 2 FORM_PRINTED = FALSE;
: 1307      1292 2
: 1308      1293 2
: 1309      1294 2 ! Initialize the item list.
```

```
BBLOCK[184],
VOLATILE VECTOR[255,BYTE],
VOLATILE WORD,
VOLATILE BBLOCK[4],
VOLATILE,
VOLATILE,
VOLATILE,
VOLATILE,
VOLATILE,
VOLATILE,
VOLATILE VECTOR[31,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE VECTOR[255,BYTE],
VOLATILE WORD,
VOLATILE VECTOR[31,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE VECTOR[255,BYTE],
VOLATILE WORD,
VECTOR[2],
```



```
1310 1295 2 1
1311 1296 2 1
1312 1297 2 1
1313 1298 2 1
1314 1299 2 1
1315 1300 2 1
1316 1301 2 1
1317 1302 2 1
1318 1303 2 1
1319 1304 2 1
1320 1305 2 1
1321 1306 2 1
1322 1307 2 1
1323 1308 2 1
1324 1309 2 1
1325 1310 2 1
1326 1311 2 1
1327 1312 2 1
1328 1313 2 1
1329 1314 2 1
1330 1315 2 1
1331 1316 2 1
1332 1317 2 1
1333 1318 2 1
1334 1319 2 1
1335 1320 2 1
1336 1321 2 1
1337 1322 2 1
1338 1323 2 1
1339 1324 2 1
1340 1325 2 1
1341 1326 2 1
1342 1327 2 1
1343 1328 2 1
1344 1329 2 1
1345 1330 2 1
1346 1331 2 1
1347 1332 2 1
1348 1333 2 1
1349 1334 2 1
1350 1335 2 1
1351 1336 2 1
1352 1337 2 1
1353 1338 2 1
1354 1339 2 1
1355 1340 2 1
1356 1341 2 1
1357 1342 2 1
1358 1343 2 1
1359 1344 2 1
1360 1345 2 1
1361 1346 2 1
1362 1347 2 1
1363 1348 2 1
1364 1349 2 1
1365 1350 2 1
1366 1351 2 1

ITMLST[0.0,16.0] = .P2_DESC[DSCSW_LENGTH];
ITMLST[2.0,16.0] = QUI$ SEARCH_NAME;
ITMLST[4.0,32.0] = .P2_DESC[DSCSA_POINTER];
ITMLST[8.0,32.0] = 0;

ITMLST[12.0,16.0] = 4;
ITMLST[14.0,16.0] = QUI$ SEARCH_FLAGS;
ITMLST[16.0,32.0] = UPLIT(QUI$M_SEARCH_WILDCARD);
ITMLST[20.0,32.0] = 0;

ITMLST[24.0,16.0] = %ALLOCATION(FORM_NAME);
ITMLST[26.0,16.0] = QUI$ FORM_NAME;
ITMLST[28.0,32.0] = FORM_NAME;
ITMLST[32.0,32.0] = FORM_NAME_LENGTH;

ITMLST[36.0,16.0] = %ALLOCATION(FORM_NUMBER);
ITMLST[38.0,16.0] = QUI$ FORM_NUMBER;
ITMLST[40.0,32.0] = FORM_NUMBER;
ITMLST[44.0,32.0] = 0;

ITMLST[48.0,16.0] = %ALLOCATION(FORM_DESCRIPTION);
ITMLST[50.0,16.0] = QUI$ FORM_DESCRIPTION;
ITMLST[52.0,32.0] = FORM_DESCRIPTION;
ITMLST[56.0,32.0] = FORM_DESCRIPTION_LENGTH;

ITMLST[60.0,32.0] = 0;

IF .FLAGS[V_FULL]
THEN
BEGIN
ITMLST[60.0,16.0] = %ALLOCATION(FORM_FLAGS);
ITMLST[62.0,16.0] = QUI$ FORM_FLAGS;
ITMLST[64.0,32.0] = FORM_FLAGS;
ITMLST[68.0,32.0] = 0;

ITMLST[72.0,16.0] = %ALLOCATION(FORM_LENGTH);
ITMLST[74.0,16.0] = QUI$ FORM_LENGTH;
ITMLST[76.0,32.0] = FORM_LENGTH;
ITMLST[80.0,32.0] = 0;

ITMLST[84.0,16.0] = %ALLOCATION(FORM_MARGIN_BOTTOM);
ITMLST[86.0,16.0] = QUI$ FORM_MARGIN_BOTTOM;
ITMLST[88.0,32.0] = FORM_MARGIN_BOTTOM;
ITMLST[92.0,32.0] = 0;

ITMLST[96.0,16.0] = %ALLOCATION(FORM_MARGIN_LEFT);
ITMLST[98.0,16.0] = QUI$ FORM_MARGIN_LEFT;
ITMLST[100.0,32.0] = FORM_MARGIN_LEFT;
ITMLST[104.0,32.0] = 0;

ITMLST[108.0,16.0] = %ALLOCATION(FORM_MARGIN_RIGHT);
ITMLST[110.0,16.0] = QUI$ FORM_MARGIN_RIGHT;
ITMLST[112.0,32.0] = FORM_MARGIN_RIGHT;
ITMLST[116.0,32.0] = 0;
```

```
: 1367      1352  3      ITMLST[120,0,16,0] = %ALLOCATION(FORM_MARGIN_TOP);
: 1368      1353  3      ITMLST[122,0,16,0] = QUI$ FORM_MARGIN_TOP;
: 1369      1354  3      ITMLST[124,0,32,0] = FORM_MARGIN_TOP;
: 1370      1355  3      ITMLST[128,0,32,0] = 0;
: 1371      1356  3
: 1372      1357  3      ITMLST[132,0,16,0] = %ALLOCATION(FORM_SETUP_MODULES);
: 1373      1358  3      ITMLST[134,0,16,0] = QUI$ FORM_SETUP_MODULES;
: 1374      1359  3      ITMLST[136,0,32,0] = FORM_SETUP_MODULES;
: 1375      1360  3      ITMLST[140,0,32,0] = FORM_SETUP_MODULES_LENGTH;
: 1376      1361  3
: 1377      1362  3      ITMLST[144,0,16,0] = %ALLOCATION(FORM_STOCK);
: 1378      1363  3      ITMLST[146,0,16,0] = QUI$ FORM_STOCK;
: 1379      1364  3      ITMLST[148,0,32,0] = FORM_STOCK;
: 1380      1365  3      ITMLST[152,0,32,0] = FORM_STOCK_LENGTH;
: 1381      1366  3
: 1382      1367  3      ITMLST[156,0,16,0] = %ALLOCATION(FORM_WIDTH);
: 1383      1368  3      ITMLST[158,0,16,0] = QUI$ FORM_WIDTH;
: 1384      1369  3      ITMLST[160,0,32,0] = FORM_WIDTH;
: 1385      1370  3      ITMLST[164,0,32,0] = 0;
: 1386      1371  3
: 1387      1372  3      ITMLST[168,0,16,0] = %ALLOCATION(PAGE_SETUP_MODULES);
: 1388      1373  3      ITMLST[170,0,16,0] = QUI$ PAGE_SETUP_MODULES;
: 1389      1374  3      ITMLST[172,0,32,0] = PAGE_SETUP_MODULES;
: 1390      1375  3      ITMLST[176,0,32,0] = PAGE_SETUP_MODULES_LENGTH;
: 1391      1376  3
: 1392      1377  3      ITMLST[180,0,32,0] = 0;
: 1393      1378  3      END;
: 1394      1379  3
: 1395      1380  3
: 1396      1381  3      ! Loop to get all forms.
: 1397      1382  3      !
: 1398      1383  3      WHILE TRUE DO
: 1399      1384  3      BEGIN
: 1400      1385  3      STATUS = $GETQUIW(
: 1401      1386  3      FUNC=QUI$ DISPLAY_FORM,
: 1402      1387  3      ITMLST=ITMLST,
: 1403      1388  3      IOSB=IOSB);
: 1404      1389  3      IF .STATUS THEN STATUS = .IOSB;
: 1405      1390  3
: 1406      1391  3
: 1407      1392  3      FLAGS[V TRUNCATE] = TRUE;
: 1408      1393  3      IF TESTBITCS(FORM_PRINTED)
: 1409      1394  3      THEN
: 1410      1395  4      BEGIN
: 1411      1396  4      IF NOT .STATUS
: 1412      1397  4      THEN
: 1413      1398  5      BEGIN
: 1414      1399  5      SIGNAL(.STATUS);
: 1415      1400  5      EXITLOOP;
: 1416      1401  4      END;
: 1417      1402  4      FAOL('Form name
: 1418      1403  4      FAOL('-----
: 1419      1404  4      END
: 1420      1405  3      ELSE
: 1421      1406  4      BEGIN
: 1422      1407  4      IF NOT .STATUS THEN EXITLOOP;
: 1423      1408  3      END;
```

Number	Description'
-----	-----'


```
: 1424      1409      3
: 1425      1410      3
: 1426      1411      3
: 1427      1412      3
: 1428      1413      3
: 1429      1414      3
: 1430      1415      3
: 1431      1416      3
: 1432      1417      3
: 1433      1418      3
: 1434      1419      3
: 1435      1420      4
: 1436      1421      4
: 1437      1422      4
: 1438      1423      4
: 1439      1424      4
: 1440      1425      4
: 1441      1426      4
: 1442      1427      4
: 1443      1428      4
: 1444      1429      4
: 1445      1430      4
: 1446      1431      4
: 1447      1432      4
: 1448      1433      4
: 1449      1434      4
: 1450      1435      4
: 1451      1436      4
: 1452      1437      4
: 1453      1438      5
: 1454      1439      5
: 1455      1440      5
: 1456      1441      5
: 1457      1442      4
: 1458      1443      4
: 1459      1444      4
: 1460      1445      4
: 1461      1446      4
: 1462      1447      5
: 1463      1448      5
: 1464      1449      5
: 1465      1450      5
: 1466      1451      4
: 1467      1452      4
: 1468      1453      4
: 1469      1454      4
: 1470      1455      4
: 1471      1456      5
: 1472      1457      5
: 1473      1458      5
: 1474      1459      4
: 1475      1460      4
: 1476      1461      4
: 1477      1462      4
: 1478      1463      4
: 1479      1464      4
: 1480      1465      4

P 1411      FAOL ('!31AF !9UL !AF',
P 1412      .FORM_NAME_LENGTH, FORM_NAME,
P 1413      .FORM_NUMBER,
P 1414      .FORM_DESCRIPTION_LENGTH, FORM_DESCRIPTION);
P 1415      FLAGS[V_TRUNCATE] = FALSE;
P 1416
P 1417      IF .FLAGS[V_FULL]
P 1418      THEN
P 1419      BEGIN
P 1420      FLAGS[B_LEFT_MARGIN] = 4;
P 1421      FLAGS[B_PENDING_SPACES] = 4;
P 1422
P 1423      FAO ('/LENGTH=!UL', .FORM_LENGTH);
P 1424      FLAGS[B_PENDING_SPACES] = 1;
P 1425
P 1426      FORMAT MARGINS(
P 1427      .FORM_MARGIN_BOTTOM,
P 1428      .FORM_MARGIN_LEFT,
P 1429      .FORM_MARGIN_RIGHT,
P 1430      .FORM_MARGIN_TOP);
P 1431
P 1432      IF .PAGE_SETUP_MODULES_LENGTH NEQ 0
P 1433      THEN
P 1434      BEGIN
P 1435      FAO ('/PAGE_SETUP=(!AF)',
P 1436      .PAGE_SETUP_MODULES_LENGTH, PAGE_SETUP_MODULES);
P 1437      FLAGS[B_PENDING_SPACES] = 1;
P 1438      END;
P 1439
P 1440      IF .FORM_SETUP_MODULES_LENGTH NEQ 0
P 1441      THEN
P 1442      BEGIN
P 1443      FAO ('/SETUP=(!AF)',
P 1444      .FORM_SETUP_MODULES_LENGTH, FORM_SETUP_MODULES);
P 1445      FLAGS[B_PENDING_SPACES] = 1;
P 1446      END;
P 1447
P 1448      IF .FORM_FLAGS[QUI$V_FORM_SHEET_FEED]
P 1449      THEN
P 1450      BEGIN
P 1451      FAO ('/SHEET_FEED');
P 1452      FLAGS[B_PENDING_SPACES] = 1;
P 1453      END;
P 1454
P 1455      FAO ('/STOCK=!AF',
P 1456      .FORM_STOCK_LENGTH, FORM_STOCK);
P 1457      FLAGS[B_PENDING_SPACES] = 1;
```

END:

Nu\

--\

00FC 00000 SHOW_FORMS:

	57	0000V	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	1238
	56	0000V	CF	9E	00007	MOVAB	EOL, R7	
	55	FF08	CF	9E	0000C	MOVAB	FA0, R6	
	54	0000	CF	9E	00011	MOVAB	P.ACI, R5	
	5E	FBCC	CE	9E	00016	MOVAB	FLAGS+7, R4	
			53	D4	0001B	MOVAB	-1076(SP), SP	
			C4	B0	0001D	CLRL	FORM PRINTED	1291
FF48	CD	008D	8F	9B	00024	MOVW	P2_DESC, ITMLST	1296
FF4A	CD	4D	C4	D0	0002A	MOVZBW	#77, ITMLST+2	1297
FF4C	CD	0091	CD	D4	00031	MOVL	P2_DESC+4, ITMLST+4	1299
		FF50	8F	D0	00035	CLRL	ITMLST+8	1299
FF54	CD	004C0004	65	9E	0003E	MOVL	#4980740, ITMLST+12	1301
FF58	CD		CD	D4	00043	MOVAB	P.ACI, ITMLST+16	1303
		FF5C	8F	D0	00047	CLRL	ITMLST+20	1304
FF60	CD	001F001F	CD	9E	00050	MOVL	#2031647, ITMLST+24	1306
FF64	CD	FE0C	CD	9E	00057	MOVAB	FORM_NAME, ITMLST+28	1308
FF68	CD	FE0A	8F	D0	0005E	MOVAB	FORM_NAME_LENGTH, ITMLST+32	1309
FF6C	CD	00200004	CD	9E	00067	MOVL	#2097156, ITMLST+36	1311
FF70	CD	FE04	CD	D4	0006E	MOVAB	FORM_NUMBER, ITMLST+40	1313
		FF74	8F	D0	00072	CLRL	ITMLST+44	1314
FF78	CD	001800FF	CD	9E	0007B	MOVL	#1573119, ITMLST+48	1316
FF7C	CD	FE48	CD	9E	00082	MOVAB	FORM_DESCRIPTION, ITMLST+52	1318
80	AD	FE46	AD	D4	00088	MOVAB	FORM_DESCRIPTION_LENGTH, ITMLST+56	1319
		84	01	E0	0008B	CLRL	ITMLST+60	1321
03	F9	A4	00B4	31	00090	BBS	#1, FLAGS, 1\$	1324
			8F	D0	00093	BRW	2\$	
84	AD	00190004	CD	9E	0009B	MOVL	#1638404, ITMLST+60	1327
88	AD	FE40	AD	D4	000A1	MOVAB	FORM_FLAGS, ITMLST+64	1329
		8C	8F	D0	000A4	CLRL	ITMLST+68	1330
90	AD	001A0004	CD	9E	000AC	MOVL	#1703940, ITMLST+72	1332
94	AD	FE3C	AD	D4	000B2	MOVAB	FORM_LENGTH, ITMLST+76	1334
		98	8F	D0	000B5	CLRL	ITMLST+80	1335
9C	AD	001B0004	CD	9E	000BD	MOVL	#1769476, ITMLST+84	1337
A0	AD	FE38	AD	D4	000C3	MOVAB	FORM_MARGIN_BOTTOM, ITMLST+88	1339
		A4	8F	D0	000C6	CLRL	ITMLST+92	1340
A8	AD	001C0004	CD	9E	000CE	MOVL	#1835012, ITMLST+96	1342
AC	AD	FE34	AD	D4	000D4	MOVAB	FORM_MARGIN_LEFT, ITMLST+100	1344
		B0	8F	D0	000D7	CLRL	ITMLST+104	1345
B4	AD	001D0004	CD	9E	000DF	MOVL	#1900548, ITMLST+108	1347
B8	AD	FE30	AD	D4	000E5	MOVAB	FORM_MARGIN_RIGHT, ITMLST+112	1349
		BC	8F	D0	000E8	CLRL	ITMLST+116	1350
C0	AD	001E0004	CD	9E	000F0	MOVL	#1966084, ITMLST+120	1352
C4	AD	FE2C	AD	D4	000F6	MOVAB	FORM_MARGIN_TOP, ITMLST+124	1354
		C8	8F	D0	000F9	CLRL	ITMLST+128	1355
CC	AD	002100FF	CE	9E	00101	MOVL	#2162943, ITMLST+132	1357
D0	AD	0138	CE	9E	00107	MOVAB	FORM_SETUP_MODULES, ITMLST+136	1359
D4	AD	0136	8F	D0	0010D	MOVAB	FORM_SETUP_MODULES_LENGTH, ITMLST+140	1360
D8	AD	0022001F	CE	9E	00115	MOVL	#2228255, ITMLST+144	1362
DC	AD	0114	CE	9E	0011B	MOVAB	FORM_STOCK, ITMLST+148	1364
E0	AD	0112	8F	D0	00121	MOVAB	FORM_STOCK_LENGTH, ITMLST+152	1365
E4	AD	00230004	CE	9E	00129	MOVL	#2293764, ITMLST+156	1367
E8	AD	010C	AD	D4	0012F	MOVAB	FORM_WIDTH, ITMLST+160	1369
		EC	8F	D0	00132	CLRL	ITMLST+164	1370
F0	AD	003900FF	AE	9E	0013A	MOVL	#3735807, ITMLST+168	1372
F4	AD	0C	AE	9E	0013F	MOVAB	PAGE_SETUP_MODULES, ITMLST+172	1374
F8	AD	0A	AD	D4	00144	MOVAB	PAGE_SETUP_MODULES_LENGTH, ITMLST+176	1375
		FC				CLRL	ITMLST+180	1377

			7E	7C	00147	2\$:	CLRQ	-(SP)		1388
		08	AE	9F	00149		PUSHAB	IOSB		
		FF48	CD	9F	0014C		PUSHAB	ITMLST		
		7E	04	7D	00150		MOVQ	#4, -(SP)		
			7E	D4	00153		CLRL	-(SP)		
			07	FB	00155		CALLS	#7, SYSSGETQUIW		
			50	D0	0015C		MOVL	R0, STATUS		
			52	E9	0015F		BLBC	STATUS, 3\$		1389
			6E	D0	00162		MOVL	IOSB, STATUS		
			04	88	00165	3\$:	BISB2	#4, FLAGS		1392
			52	D2	00169		MCOML	STATUS, R0		1396
			00	E2	0016C		BBSS	#0, FORM_PRINTED, 5\$		1393
			50	E9	00170		BLBC	R0, 4\$		1396
			52	DD	00173		PUSHL	STATUS		1399
			01	FB	00175		CALLS	#1, LIB\$SIGNAL		
				04	0017C		RET			1398
			04	A5	9F	0017D	4\$:	PUSHAB	P.ACJ	1402
			66	01	FB	00180		CALLS	#1, FAO	
			67	00	FB	00183		CALLS	#0, EOL	
				A5	9F	00186		PUSHAB	P.ACK	1403
			66	01	FB	00189		CALLS	#1, FAO	
			67	00	FB	0018C		CALLS	#0, EOL	
				04	11	0018F		BRB	6\$	1393
			01	50	E9	00191	5\$:	BLBC	R0, 6\$	1407
				04	00194		RET			
				CD	9F	00195	6\$:	PUSHAB	FORM_DESCRIPTION	1414
			7E	CD	3C	00199		MOVZWL	FORM_DESCRIPTION_LENGTH, -(SP)	
				CD	DD	0019E		PUSHL	FORM_NUMBER	
				CD	9F	001A2		PUSHAB	FORM_NAME	
			7E	CD	3C	001A6		MOVZWL	FORM_NAME_LENGTH, -(SP)	
				A5	9F	001AB		PUSHAB	P.ACC	
			66	06	FB	001AE		CALLS	#6, FAO	
			67	00	FB	001B1		CALLS	#0, EOL	
				04	8A	001B4		BICB2	#4, FLAGS	1415
				01	E1	001B8		BBC	#1, FLAGS, 2\$	1418
				04	90	001BD		MOVB	#4, FLAGS+4	1421
				04	90	001C1		MOVB	#4, FLAGS+7	1422
				CD	DD	001C4		PUSHL	FORM_LENGTH	1425
				C5	9F	001C8		PUSHAB	P.ACM	
			66	02	FB	001CC		CALLS	#2, FAO	
			64	01	90	001CF		MOVB	#1, FLAGS+7	1426
				CD	DD	001D2		PUSHL	FORM_MARGIN_TOP	1433
				CD	DD	001D6		PUSHL	FORM_MARGIN_RIGHT	1432
				CD	DD	001DA		PUSHL	FORM_MARGIN_LEFT	1431
				CD	DD	001DE		PUSHL	FORM_MARGIN_BOTTOM	1430
				04	FB	001E2		CALLS	#4, FORMAT MARGINS	
				AE	B5	001E7		TSTW	PAGE_SETUP_MODULES_LENGTH	1436
				11	13	001EA		BEQL	7\$	
				AE	9F	001EC		PUSHAB	PAGE_SETUP_MODULES	1440
			7E	AE	3C	001EF		MOVZWL	PAGE_SETUP_MODULES_LENGTH, -(SP)	
				C5	9F	001F3		PUSHAB	P.ACN	
			66	03	FB	001F7		CALLS	#3, FAO	
			64	01	90	001FA		MOVB	#1, FLAGS+7	1441
				CE	B5	001FD	7\$:	TSTW	FORM_SETUP_MODULES_LENGTH	1445
				13	13	00201		BEQL	8\$	
				CE	9F	00203		PUSHAB	FORM_SETUP_MODULES	1449
			7E	CE	3C	00207		MOVZWL	FORM_SETUP_MODULES_LENGTH, -(SP)	

			00A9	C5	9F	0020C	PUSHAB	P.ACO		
66				03	FB	00210	CALLS	#3, FAO		
64				01	90	00213	MOVB	#1, FLAGS+7		1450
0A			FE40	CD	E9	00216	BLBC	FORM_FLAGS, 9\$		1454
			00B6	C5	9F	0021B	PUSHAB	P.ACP		1457
66				01	FB	0021F	CALLS	#1, FAO		
64				01	90	00222	MOVB	#1, FLAGS+7		1458
			0114	CE	9F	00225	PUSHAB	FORM_STOCK		1463
7E			0116	CE	3C	00229	MOVZWL	FORM_STOCK_LENGTH, -(SP)		
			00C2	C5	9F	0022E	PUSHAB	P.ACO		
66				03	FB	00232	CALLS	#3, FAO		
64				01	90	00235	MOVB	#1, FLAGS+7		1464
0A	FE40		CD	01	E1	00238	BBC	#1, FORM_FLAGS, 10\$		1467
			00CD	C5	9F	0023E	PUSHAB	P.ACR		1470
66				01	FB	00242	CALLS	#1, FAO		
64				01	90	00245	MOVB	#1, FLAGS+7		1471
			010C	CE	DD	00248	PUSHL	FORM_WIDTH		1475
			00D7	C5	9F	0024C	PUSHAB	P.ACS		
66				02	FB	00250	CALLS	#2, FAO		
64				01	90	00253	MOVB	#1, FLAGS+7		1476
0A	FE40		CD	02	E1	00256	BBC	#2, FORM_FLAGS, 11\$		1479
			00E2	C5	9F	0025C	PUSHAB	P.ACT		1482
66				01	FB	00260	CALLS	#1, FAO		
64				01	90	00263	MOVB	#1, FLAGS+7		1483
				64	94	00266	CLRB	FLAGS+7		1487
			FD	A4	94	00268	CLRB	FLAGS+4		1488
	0000V	CF		00	FB	0026B	CALLS	#0, CEOL		1489
				FED4	31	00270	BRW	2\$		1383
				04	00	00273	RET			1492

; Routine Size: 628 bytes, Routine Base: CODE + 0B04

```
: 1509      1493 1 ROUTINE SHOW_QUEUES: NOVALUE=
: 1510      1494 1
: 1511      1495 1 ++
: 1512      1496 1
: 1513      1497 1 FUNCTIONAL DESCRIPTION:
: 1514      1498 1     This routine displays queues.
: 1515      1499 1
: 1516      1500 1 INPUT PARAMETERS:
: 1517      1501 1     NONE
: 1518      1502 1
: 1519      1503 1 IMPLICIT INPUTS:
: 1520      1504 1     NONE
: 1521      1505 1
: 1522      1506 1 OUTPUT PARAMETERS:
: 1523      1507 1     NONE
: 1524      1508 1
: 1525      1509 1 IMPLICIT OUTPUTS:
: 1526      1510 1     NONE
: 1527      1511 1
: 1528      1512 1 ROUTINE VALUE:
: 1529      1513 1     NONE
: 1530      1514 1
: 1531      1515 1 SIDE EFFECTS:
: 1532      1516 1     NONE
: 1533      1517 1
: 1534      1518 1 --
: 1535      1519 1
: 1536      1520 2 BEGIN
: 1537      1521 2 LOCAL
: 1538      1522 2     ITMLST:
: 1539      1523 2     ASSIGNED_QUEUE_NAME:
: 1540      1524 2     ASSIGNED_QUEUE_NAME_LENGTH:
: 1541      1525 2     BASE_PRIORITY:
: 1542      1526 2     CHARACTERISTICS:
: 1543      1527 2     CPU_DEFAULT:
: 1544      1528 2     CPU_LIMIT:
: 1545      1529 2     DEVICE_NAME:
: 1546      1530 2     DEVICE_NAME_LENGTH:
: 1547      1531 2     FORM_NAME:
: 1548      1532 2     FORM_NAME_LENGTH:
: 1549      1533 2     GENERIC_TARGET:
: 1550      1534 2     GENERIC_TARGET_LENGTH:
: 1551      1535 2     JOB_LIMIT:
: 1552      1536 2     JOB_RESET_MODULES:
: 1553      1537 2     JOB_RESET_MODULES_LENGTH:
: 1554      1538 2     JOB_SIZE_MAXIMUM:
: 1555      1539 2     JOB_SIZE_MINIMUM:
: 1556      1540 2     LIBRARY_SPECIFICATION:
: 1557      1541 2     LIBRARY_SPECIFICATION_LENGTH:
: 1558      1542 2     OWNER_UIC:
: 1559      1543 2     PROCESSOR:
: 1560      1544 2     PROCESSOR_LENGTH:
: 1561      1545 2     PROTECTION:
: 1562      1546 2     QUEUE_FLAGS:
: 1563      1547 2     QUEUE_NAME:
: 1564      1548 2     QUEUE_NAME_LENGTH:
: 1565      1549 2     QUEUE_STATOS:

BBLOCK[304],
VOLATILE VECTOR[39,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE BITVECTOR[128],
VOLATILE,
VOLATILE,
VOLATILE VECTOR[31,BYTE],
VOLATILE WORD,
VOLATILE VECTOR[31,BYTE],
VOLATILE WORD,
VOLATILE VECTOR[255,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE VECTOR[255,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE VECTOR[39,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE VECTOR[39,BYTE],
VOLATILE WORD,
VOLATILE,
VOLATILE BBLOCK[4],
VOLATILE VECTOR[31,BYTE],
VOLATILE WORD,
VOLATILE BBLOCK[4],
```



```

: 1566      1550 2      SEARCH_FLAGS:      VOLATILE BBLOCK[4],
: 1567      1551 2      SCSNODE_NAME:      VOLATILE VECTOR[6,BYTE],
: 1568      1552 2      SCSNODE_NAME_LENGTH:  VOLATILE WORD,
: 1569      1553 2      WSDEFAULT:      VOLATILE,
: 1570      1554 2      WSEXTENT:      VOLATILE,
: 1571      1555 2      WSQUOTA:      VOLATILE,
: 1572      1556 2      STATUS,
: 1573      1557 2      IOSB:      VECTOR[2],
: 1574      1558 2      QUEUE_PRINTED,
: 1575      1559 2      TYPE;
: 1576      1560 2
: 1577      1561 2
: 1578      1562 2      QUEUE_PRINTED = FALSE;
: 1579      1563 2
: 1580      1564 2
: 1581      1565 2      SEARCH_FLAGS = QUI$M SEARCH_WILDCARD;
: 1582      1566 2      IF CLIP$PRESENT($DESCRIPTOR('BATCH'))
: 1583      1567 2          THEN SEARCH_FLAGS[QUI$V_SEARCH_BATCH] = TRUE;
: 1584      1568 2      IF CLIP$PRESENT($DESCRIPTOR('DEVICE'))
: 1585      1569 2          THEN SEARCH_FLAGS[QUI$V_SEARCH_SYMBIONT] = TRUE;
: 1586      1570 2
: 1587      1571 2
: 1588      1572 2      ! Initialize item list.
: 1589      1573 2      !
: 1590      1574 2      ITMLST[0,0,16,0] = .P2_DESC[DSC$W_LENGTH];
: 1591      1575 2      ITMLST[2,0,16,0] = QUI$ SEARCH_NAME;
: 1592      1576 2      ITMLST[4,0,32,0] = .P2_DESC[DSC$A_POINTER];
: 1593      1577 2      ITMLST[8,0,32,0] = 0;
: 1594      1578 2
: 1595      1579 2      ITMLST[12,0,16,0] = %ALLOCATION(SEARCH_FLAGS);
: 1596      1580 2      ITMLST[14,0,16,0] = QUI$ SEARCH_FLAGS;
: 1597      1581 2      ITMLST[16,0,32,0] = SEARCH_FLAGS;
: 1598      1582 2      ITMLST[20,0,32,0] = 0;
: 1599      1583 2
: 1600      1584 2      ITMLST[24,0,16,0] = %ALLOCATION(ASSIGNED_QUEUE_NAME);
: 1601      1585 2      ITMLST[26,0,16,0] = QUI$ ASSIGNED_QUEUE_NAME;
: 1602      1586 2      ITMLST[28,0,32,0] = ASSIGNED_QUEUE_NAME;
: 1603      1587 2      ITMLST[32,0,32,0] = ASSIGNED_QUEUE_NAME_LENGTH;
: 1604      1588 2
: 1605      1589 2      ITMLST[36,0,16,0] = %ALLOCATION(DEVICE_NAME);
: 1606      1590 2      ITMLST[38,0,16,0] = QUI$ DEVICE_NAME;
: 1607      1591 2      ITMLST[40,0,32,0] = DEVICE_NAME;
: 1608      1592 2      ITMLST[44,0,32,0] = DEVICE_NAME_LENGTH;
: 1609      1593 2
: 1610      1594 2      ITMLST[48,0,16,0] = %ALLOCATION(PROCESSOR);
: 1611      1595 2      ITMLST[50,0,16,0] = QUI$ PROCESSOR;
: 1612      1596 2      ITMLST[52,0,32,0] = PROCESSOR;
: 1613      1597 2      ITMLST[56,0,32,0] = PROCESSOR_LENGTH;
: 1614      1598 2
: 1615      1599 2      ITMLST[60,0,16,0] = %ALLOCATION(QUEUE_FLAGS);
: 1616      1600 2      ITMLST[62,0,16,0] = QUI$ QUEUE_FLAGS;
: 1617      1601 2      ITMLST[64,0,32,0] = QUEUE_FLAGS;
: 1618      1602 2      ITMLST[68,0,32,0] = 0;
: 1619      1603 2
: 1620      1604 2      ITMLST[72,0,16,0] = %ALLOCATION(QUEUE_NAME);
: 1621      1605 2      ITMLST[74,0,16,0] = QUI$ QUEUE_NAME;
: 1622      1606 2      ITMLST[76,0,32,0] = QUEUE_NAME;

```



```
: 1623      1607  2 ITMLST[80,0,32,0] = QUEUE_NAME_LENGTH;
: 1624      1608  2
: 1625      1609  2 ITMLST[84,0,16,0] = %ALLOCATION(QUEUE_STATUS);
: 1626      1610  2 ITMLST[86,0,16,0] = QUI$ QUEUE_STATUS;
: 1627      1611  2 ITMLST[88,0,32,0] = QUEUE_STATUS;
: 1628      1612  2 ITMLST[92,0,32,0] = 0;
: 1629      1613  2
: 1630      1614  2 ITMLST[96,0,16,0] = %ALLOCATION(SCSNODE_NAME);
: 1631      1615  2 ITMLST[98,0,16,0] = QUI$ SCSNODE_NAME;
: 1632      1616  2 ITMLST[100,0,32,0] = SCSNODE_NAME;
: 1633      1617  2 ITMLST[104,0,32,0] = SCSNODE_NAME_LENGTH;
: 1634      1618  2
: 1635      1619  2 ITMLST[108,0,32,0] = 0;
: 1636      1620  2
: 1637      1621  2
: 1638      1622  2 IF .FLAGS[V_FULL]
: 1639      1623  2 THEN
: 1640      1624  2 BEGIN
: 1641      1625  2 ITMLST[108,0,16,0] = %ALLOCATION(BASE_PRIORITY);
: 1642      1626  2 ITMLST[110,0,16,0] = QUI$ BASE_PRIORITY;
: 1643      1627  2 ITMLST[112,0,32,0] = BASE_PRIORITY;
: 1644      1628  2 ITMLST[116,0,32,0] = 0;
: 1645      1629  2
: 1646      1630  2 ITMLST[120,0,16,0] = %ALLOCATION(CCHARACTERISTICS);
: 1647      1631  2 ITMLST[122,0,16,0] = QUI$ CHARACTERISTICS;
: 1648      1632  2 ITMLST[124,0,32,0] = CHARACTERISTICS;
: 1649      1633  2 ITMLST[128,0,32,0] = 0;
: 1650      1634  2
: 1651      1635  2 ITMLST[132,0,16,0] = %ALLOCATION(CPU_DEFAULT);
: 1652      1636  2 ITMLST[134,0,16,0] = QUI$ CPU_DEFAULT;
: 1653      1637  2 ITMLST[136,0,32,0] = CPU_DEFAULT;
: 1654      1638  2 ITMLST[140,0,32,0] = 0;
: 1655      1639  2
: 1656      1640  2 ITMLST[144,0,16,0] = %ALLOCATION(CPU_LIMIT);
: 1657      1641  2 ITMLST[146,0,16,0] = QUI$ CPU_LIMIT;
: 1658      1642  2 ITMLST[148,0,32,0] = CPU_LIMIT;
: 1659      1643  2 ITMLST[152,0,32,0] = 0;
: 1660      1644  2
: 1661      1645  2 ITMLST[156,0,16,0] = %ALLOCATION(FORM_NAME);
: 1662      1646  2 ITMLST[158,0,16,0] = QUI$ FORM_NAME;
: 1663      1647  2 ITMLST[160,0,32,0] = FORM_NAME;
: 1664      1648  2 ITMLST[164,0,32,0] = FORM_NAME_LENGTH;
: 1665      1649  2
: 1666      1650  2 ITMLST[168,0,16,0] = %ALLOCATION(GENERIC_TARGET);
: 1667      1651  2 ITMLST[170,0,16,0] = QUI$ GENERIC_TARGET;
: 1668      1652  2 ITMLST[172,0,32,0] = GENERIC_TARGET;
: 1669      1653  2 ITMLST[176,0,32,0] = GENERIC_TARGET_LENGTH;
: 1670      1654  2
: 1671      1655  2 ITMLST[180,0,16,0] = %ALLOCATION(JOB_LIMIT);
: 1672      1656  2 ITMLST[182,0,16,0] = QUI$ JOB_LIMIT;
: 1673      1657  2 ITMLST[184,0,32,0] = JOB_LIMIT;
: 1674      1658  2 ITMLST[188,0,32,0] = 0;
: 1675      1659  2
: 1676      1660  2 ITMLST[192,0,16,0] = %ALLOCATION(JOB_RESET_MODULES);
: 1677      1661  2 ITMLST[194,0,16,0] = QUI$ JOB_RESET_MODULES;
: 1678      1662  2 ITMLST[196,0,32,0] = JOB_RESET_MODULES;
: 1679      1663  2 ITMLST[200,0,32,0] = JOB_RESET_MODULES_LENGTH;
```



```
: 1680      1664      3
: 1681      1665      3      ITMLST[204,0,16,0] = %ALLOCATION(JOB_SIZE_MAXIMUM);
: 1682      1666      3      ITMLST[206,0,16,0] = QUI$ JOB_SIZE_MAXIMUM;
: 1683      1667      3      ITMLST[208,0,32,0] = JOB_SIZE_MAXIMUM;
: 1684      1668      3      ITMLST[212,0,32,0] = 0;
: 1685      1669      3
: 1686      1670      3      ITMLST[216,0,16,0] = %ALLOCATION(JOB_SIZE_MINIMUM);
: 1687      1671      3      ITMLST[218,0,16,0] = QUI$ JOB_SIZE_MINIMUM;
: 1688      1672      3      ITMLST[220,0,32,0] = JOB_SIZE_MINIMUM;
: 1689      1673      3      ITMLST[224,0,32,0] = 0;
: 1690      1674      3
: 1691      1675      3      ITMLST[228,0,16,0] = %ALLOCATION(LIBRARY_SPECIFICATION);
: 1692      1676      3      ITMLST[230,0,16,0] = QUI$ LIBRARY_SPECIFICATION;
: 1693      1677      3      ITMLST[232,0,32,0] = LIBRARY_SPECIFICATION;
: 1694      1678      3      ITMLST[236,0,32,0] = LIBRARY_SPECIFICATION_LENGTH;
: 1695      1679      3
: 1696      1680      3      ITMLST[240,0,16,0] = %ALLOCATION(OWNER_UIC);
: 1697      1681      3      ITMLST[242,0,16,0] = QUI$ OWNER_UIC;
: 1698      1682      3      ITMLST[244,0,32,0] = OWNER_UIC;
: 1699      1683      3      ITMLST[248,0,32,0] = 0;
: 1700      1684      3
: 1701      1685      3      ITMLST[252,0,16,0] = %ALLOCATION(PROTECTION);
: 1702      1686      3      ITMLST[254,0,16,0] = QUI$ PROTECTION;
: 1703      1687      3      ITMLST[256,0,32,0] = PROTECTION;
: 1704      1688      3      ITMLST[260,0,32,0] = 0;
: 1705      1689      3
: 1706      1690      3      ITMLST[264,0,16,0] = %ALLOCATION(WSDEFAULT);
: 1707      1691      3      ITMLST[266,0,16,0] = QUI$ WSDEFAULT;
: 1708      1692      3      ITMLST[268,0,32,0] = WSDEFAULT;
: 1709      1693      3      ITMLST[272,0,32,0] = 0;
: 1710      1694      3
: 1711      1695      3      ITMLST[276,0,16,0] = %ALLOCATION(WSEXTENT);
: 1712      1696      3      ITMLST[278,0,16,0] = QUI$ WSEXTENT;
: 1713      1697      3      ITMLST[280,0,32,0] = WSEXTENT;
: 1714      1698      3      ITMLST[284,0,32,0] = 0;
: 1715      1699      3
: 1716      1700      3      ITMLST[288,0,16,0] = %ALLOCATION(WSQUOTA);
: 1717      1701      3      ITMLST[290,0,16,0] = QUI$ WSQUOTA;
: 1718      1702      3      ITMLST[292,0,32,0] = WSQUOTA;
: 1719      1703      3      ITMLST[296,0,32,0] = 0;
: 1720      1704      3
: 1721      1705      3      ITMLST[300,0,32,0] = 0;
: 1722      1706      3      END;
: 1723      1707      3
: 1724      1708      3
: 1725      1709      3      ! Loop to get all queues.
: 1726      1710      3      !
: 1727      1711      3      WHILE TRUE DO
: 1728      1712      3      BEGIN
: 1729      1713      3      STATUS = $GETQUIW(
P      1714      3      FUNC=QUI$ DISPLAY_QUEUE,
P      1715      3      ITMLST=ITMLST,
P      1716      3      IOSB=IOSB);
: 1730      1717      3      IF .STATUS THEN STATUS = .IOSB;
: 1731      1718      3
: 1732      1719      3
: 1733      1720      3      IF TESTBITCS(QUEUE_PRINTED)
: 1734
: 1735
: 1736
```

```
: 1737      1721 3      THEN
: 1738      1722 4      BEGIN
: 1739      1723 4      IF NOT .STATUS
: 1740      1724 4      THEN
: 1741      1725 5      BEGIN
: 1742      1726 5      SIGNAL(.STATUS);
: 1743      1727 5      EXITLOOP;
: 1744      1728 4      END;
: 1745      1729 4      END
: 1746      1730 3      ELSE
: 1747      1731 4      BEGIN
: 1748      1732 4      IF NOT .STATUS THEN EXITLOOP;
: 1749      1733 4      END;
: 1750      1734 3
: 1751      1735 3
: 1752      1736 3      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 1753      1737 3      THEN
: 1754      1738 3      IF .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1755      1739 3      THEN
: 1756      1740 3      TYPE = UPLIT BYTE (%ASCIC 'Generic batch')
: 1757      1741 3      ELSE
: 1758      1742 3      TYPE = UPLIT BYTE (%ASCIC 'Batch')
: 1759      1743 3      ELSE
: 1760      1744 3      IF .ASSIGNED_QUEUE_NAME_LENGTH NEQ 0
: 1761      1745 3      THEN
: 1762      1746 3      TYPE = UPLIT BYTE (%ASCIC 'Logical')
: 1763      1747 3      ELSE
: 1764      1748 3      IF .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1765      1749 3      THEN
: 1766      1750 3      IF .PROCESSOR_LENGTH NEQ 0
: 1767      1751 3      THEN
: 1768      1752 3      TYPE = UPLIT BYTE (%ASCIC 'Generic server')
: 1769      1753 3      ELSE IF .QUEUE_FLAGS[QUI$V_QUEUE_TERMINAL]
: 1770      1754 3      THEN
: 1771      1755 3      TYPE = UPLIT BYTE (%ASCIC 'Generic terminal')
: 1772      1756 3      ELSE
: 1773      1757 3      TYPE = UPLIT BYTE (%ASCIC 'Generic printer')
: 1774      1758 3      ELSE
: 1775      1759 3      IF .QUEUE_STATUS[QUI$V_QUEUE_SERVER]
: 1776      1760 3      THEN
: 1777      1761 3      TYPE = UPLIT BYTE (%ASCIC 'Server')
: 1778      1762 3      ELSE IF .QUEUE_FLAGS[QUI$V_QUEUE_TERMINAL]
: 1779      1763 3      THEN
: 1780      1764 3      IF .QUEUE_STATUS[QUI$V_QUEUE_REMOTE]
: 1781      1765 3      THEN
: 1782      1766 3      TYPE = UPLIT BYTE (%ASCIC 'Remote terminal')
: 1783      1767 3      ELSE
: 1784      1768 3      TYPE = UPLIT BYTE (%ASCIC 'Terminal')
: 1785      1769 3      ELSE
: 1786      1770 3      TYPE = UPLIT BYTE (%ASCIC 'Printer');
: 1787      1771 3
: 1788      1772 3
: 1789      1773 3      FLAGS[V_TRUNCATE] = TRUE;
: 1790      1774 3      FAO_('!AC queue !AF', .TYPE, .QUEUE_NAME_LENGTH, QUEUE_NAME);
: 1791      1775 3
: 1792      1776 3
: 1793      1777 3      IF .QUEUE_STATUS[QUI$V_QUEUE_ALIGNING]
```



```
: 1794      1778 3      THEN FAO (' aligning');
: 1795      1779 3      IF .QUEUE STATUS[QUI$V QUEUE_OPERATOR_REQUEST]
: 1796      1780 3      THEN FAO (' operator service');
: 1797      1781 3      IF .QUEUE STATUS[QUI$V QUEUE_PAUSED]
: 1798      1782 3      THEN FAO (' paused');
: 1799      1783 3      IF .QUEUE STATUS[QUI$V QUEUE_PAUSING]
: 1800      1784 3      THEN FAO (' pausing');
: 1801      1785 3      IF .QUEUE STATUS[QUI$V QUEUE_RESETTING]
: 1802      1786 3      THEN FAO (' resetting');
: 1803      1787 3      IF NOT .QUEUE STATUS[QUI$V QUEUE_ALIGNING]
: 1804      1788 3      AND .QUEUE STATUS[QUI$V QUEUE_RESUMING]
: 1805      1789 3      THEN FAO (' resuming');
: 1806      1790 3      IF .QUEUE STATUS[QUI$V QUEUE_STALLED]
: 1807      1791 3      THEN FAO (' stalled');
: 1808      1792 3      IF .QUEUE STATUS[QUI$V QUEUE_STARTING]
: 1809      1793 3      THEN FAO (' starting');
: 1810      1794 3      IF .QUEUE STATUS[QUI$V QUEUE_STOPPED]
: 1811      1795 3      THEN
: 1812      1796 3          IF .QUEUE STATUS[QUI$V QUEUE_IDLE]
: 1813      1797 3              THEN FAO (' stopped');
: 1814      1798 3              ELSE FAO (' stop pending');
: 1815      1799 3      IF .QUEUE STATUS[QUI$V QUEUE_STOPPING]
: 1816      1800 3      THEN FAO (' stopping');
: 1817      1801 3      IF .QUEUE STATUS[QUI$V QUEUE_UNAVAILABLE]
: 1818      1802 3      THEN FAO (' device unavailable');
: 1819      1803 3
: 1820      1804 3
: 1821      1805 3      IF .ASSIGNED_QUEUE_NAME_LENGTH NEQ 0
: 1822      1806 3      THEN
: 1823      1807 3          FAO (' assigned to !AF',
: 1824      1808 3              .ASSIGNED_QUEUE_NAME_LENGTH, ASSIGNED_QUEUE_NAME);
: 1825      1809 3
: 1826      1810 3
: 1827      1811 3      IF .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
: 1828      1812 3      AND NOT .QUEUE_FLAGS[QUI$V QUEUE_GENERIC]
: 1829      1813 3      THEN
: 1830      1814 3          BEGIN
: 1831      1815 3              IF .SCSNODE_NAME_LENGTH NEQ 0 OR .DEVICE_NAME_LENGTH NEQ 0
: 1832      1816 3              THEN
: 1833      1817 3                  FAO (' on ');
: 1834      1818 3
: 1835      1819 3              IF .SCSNODE_NAME_LENGTH NEQ 0
: 1836      1820 3              THEN
: 1837      1821 3                  FAO (' !AF::', .SCSNODE_NAME_LENGTH, SCSNODE_NAME);
: 1838      1822 3
: 1839      1823 3              IF .DEVICE_NAME_LENGTH NEQ 0
: 1840      1824 3              THEN
: 1841      1825 3                  FAO (' !AF', .DEVICE_NAME_LENGTH, DEVICE_NAME);
: 1842      1826 3
: 1843      1827 3              END;
: 1844      1828 3      FLAGS[V_TRUNCATE] = FALSE;
: 1845      1829 3
: 1846      1830 3
: 1847      1831 3      IF .FLAGS[V_FULL]
: 1848      1832 3      THEN
: 1849      1833 3          BEGIN
: 1850      1834 3
```

```
: 1851      1835  4      EOL ();
: 1852      1836  4      FLAGS[B_PENDING_SPACES] = 4;
: 1853      1837  4      FLAGS[B_LEFT_MARGIN] = 4;
: 1854      1838  4
: 1855      1839  4
: 1856      1840  4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1857      1841  4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
: 1858      1842  4      THEN
: 1859      1843  5          BEGIN
: 1860      1844  5              FAO ('/BASE_PRIORITY=!UL', .BASE_PRIORITY);
: 1861      1845  5              FLAGS[B_PENDING_SPACES] = 1;
: 1862      1846  4              END;
: 1863      1847  4
: 1864      1848  4
: 1865      1849  4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 1866      1850  4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1867      1851  4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
: 1868      1852  4      THEN
: 1869      1853  5          BEGIN
: 1870      1854  5              IF .JOB_SIZE_MINIMUM NEQ 0
: 1871      1855  5              THEN
: 1872      1856  5                  IF .JOB_SIZE_MAXIMUM NEQ 0
: 1873      1857  5                  THEN
: 1874      1858  6                      BEGIN
: 1875      1859  6                          FAO ('/BLOCK_LIMIT=(!UL,!UL)',
: 1876      1860  6                          .JOB_SIZE_MINIMUM, .JOB_SIZE_MAXIMUM);
: 1877      1861  6                          FLAGS[B_PENDING_SPACES] = 1;
: 1878      1862  6                          END
: 1879      1863  5                  ELSE
: 1880      1864  6                      BEGIN
: 1881      1865  6                          FAO ('/BLOCK_LIMIT=(!UL,"")',
: 1882      1866  6                          .JOB_SIZE_MINIMUM);
: 1883      1867  6                          FLAGS[B_PENDING_SPACES] = 1;
: 1884      1868  6                          END
: 1885      1869  5                  ELSE
: 1886      1870  5                      IF .JOB_SIZE_MAXIMUM NEQ 0
: 1887      1871  5                      THEN
: 1888      1872  6                          BEGIN
: 1889      1873  6                              FAO ('/BLOCK_LIMIT=!UL',
: 1890      1874  6                              .JOB_SIZE_MAXIMUM);
: 1891      1875  6                              FLAGS[B_PENDING_SPACES] = 1;
: 1892      1876  5                              END;
: 1893      1877  4                      END;
: 1894      1878  4
: 1895      1879  4
: 1896      1880  4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1897      1881  4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
: 1898      1882  4      THEN
: 1899      1883  4          FORMAT_CHARACTERISTICS(CCHARACTERISTICS);
: 1900      1884  4
: 1901      1885  4
: 1902      1886  4      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 1903      1887  4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1904      1888  4      THEN
: 1905      1889  5          BEGIN
: 1906      1890  5              IF .QUEUE_FLAGS[QUI$V_QUEUE_CPU_DEFAULT]
: 1907      1891  5              THEN
```



```
1908 1892 5      FORMAT CPU TIME(
1909 1893 5      UP[IT] BYTE (%ASCIC 'CPUDEFAULT'),
1910 1894 5      .CPU_DEFAULT);
1911 1895 5
1912 1896 5
1913 1897 5      IF .QUEUE_FLAGS[QUI$V_QUEUE_CPU_LIMIT]
1914 1898 5      THEN
1915 1899 5          FORMAT CPU TIME(
1916 1900 5          UP[IT] BYTE (%ASCIC 'CPUMAXIMUM'),
1917 1901 5          .CPU_LIMIT);
1918 1902 4      END;
1919 1903 4
1920 1904 4
1921 1905 4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
1922 1906 4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
1923 1907 4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
1924 1908 4      THEN
1925 1909 4          FORMAT_DEFAULT(.QUEUE_FLAGS);
1926 1910 4
1927 1911 4
1928 1912 4      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
1929 1913 4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
1930 1914 4      THEN
1931 1915 5          BEGIN
1932 1916 5              IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_SWAP]
1933 1917 5              THEN
1934 1918 6                  BEGIN
1935 1919 6                      FAO ('/DISABLE SWAPPING');
1936 1920 6                      FLAGS[B_PENDING_SPACES] = 1;
1937 1921 5                      END;
1938 1922 4          END;
1939 1923 4
1940 1924 4
1941 1925 4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
1942 1926 4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
1943 1927 4      THEN
1944 1928 5          BEGIN
1945 1929 5              IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC_SELECTION]
1946 1930 5              THEN
1947 1931 6                  BEGIN
1948 1932 6                      FAO ('/NOENABLE_GENERIC');
1949 1933 6                      FLAGS[B_PENDING_SPACES] = 1;
1950 1934 5                      END;
1951 1935 4          END;
1952 1936 4
1953 1937 4
1954 1938 4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
1955 1939 4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
1956 1940 4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
1957 1941 4      THEN
1958 1942 5          BEGIN
1959 1943 5              FAO ('/FORM=!AF', .FORM_NAME_LENGTH, FORM_NAME);
1960 1944 5              FLAGS[B_PENDING_SPACES] = 1;
1961 1945 4          END;
1962 1946 4
1963 1947 4
1964 1948 4      IF .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
```

```
: 1965      1949  4      AND .GENERIC_TARGET_LENGTH NEQ 0
: 1966      1950  4      THEN
: 1967      1951  5      BEGIN
: 1968      1952  5      LOCAL
: 1969      1953  5      ITEM_LENGTH,
: 1970      1954  5      ITEM_ADDRESS,
: 1971      1955  5      REMAINING_LENGTH,
: 1972      1956  5      REMAINING_ADDRESS;
: 1973      1957  5
: 1974      1958  5      FAO ('/GENERIC=(':
: 1975      1959  5      FLAGS[B_PENDING_SPACES] = 0;
: 1976      1960  5
: 1977      1961  5      ITEM_ADDRESS = REMAINING_ADDRESS = GENERIC_TARGET;
: 1978      1962  5      REMAINING_LENGTH = .GENERIC_TARGET_LENGTH;
: 1979      1963  5
: 1980      1964  7      WHILE ((REMAINING_ADDRESS = CH$FIND_CH(.REMAINING_LENGTH,
: 1981      1965  6      .REMAINING_ADDRESS, %C',')) NEQ 0)
: 1982      1966  6      DO BEGIN
: 1983      1967  6      ITEM_LENGTH = .REMAINING_ADDRESS - .ITEM_ADDRESS;
: 1984      1968  6      FAO T '!AF', .ITEM_LENGTH, .ITEM_ADDRESS;
: 1985      1969  6      REMAINING_LENGTH = .REMAINING_LENGTH - .ITEM_LENGTH - 1;
: 1986      1970  6      ITEM_ADDRESS = REMAINING_ADDRESS = .REMAINING_ADDRESS + 1;
: 1987      1971  5      END;
: 1988      1972  5
: 1989      1973  5      FAO ('!AF', .REMAINING_LENGTH, .ITEM_ADDRESS);
: 1990      1974  5      FAO ('')';
: 1991      1975  5      FLAGS[B_PENDING_SPACES] = 1;
: 1992      1976  4      END;
: 1993      1977  4
: 1994      1978  4
: 1995      1979  4      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 1996      1980  4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 1997      1981  4      THEN
: 1998      1982  5      BEGIN
: 1999      1983  5      FAO ('/JOB LIMIT=!UL', .JOB_LIMIT);
: 2000      1984  5      FLAGS[B_PENDING_SPACES] = 1;
: 2001      1985  4      END;
: 2002      1986  4
: 2003      1987  4
: 2004      1988  4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 2005      1989  4      AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
: 2006      1990  4      AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
: 2007      1991  4      AND .LIBRARY_SPECIFICATION_LENGTH NEQ 0
: 2008      1992  4      THEN
: 2009      1993  5      BEGIN
: 2010      1994  5      FAO ('/LIBRARY=!AF',
: 2011      1995  5      .LIBRARY_SPECIFICATION_LENGTH, LIBRARY_SPECIFICATION);
: 2012      1996  5      FLAGS[B_PENDING_SPACES] = 1;
: 2013      1997  4      END;
: 2014      1998  4
: 2015      1999  4
: 2016      2000  4      IF .QUEUE_STATUS[QUI$V_QUEUE_LOWERCASE]
: 2017      2001  4      THEN
: 2018      2002  5      BEGIN
: 2019      2003  5      FAO ('Lowercase');
: 2020      2004  5      FLAGS[B_PENDING_SPACES] = 1;
: 2021      2005  4      END;
```



```
2022 2006 4
2023 2007 4
2024 2008 4
2025 2009 4
2026 2010 4
2027 2011 4
2028 2012 4
2029 2013 4
2030 2014 4
2031 2015 5
2032 P 2016 5
2033 2017 5
2034 2018 5
2035 2019 4
2036 2020 4
2037 2021 4
2038 2022 4
2039 2023 4
2040 2024 4
2041 2025 4
2042 2026 4
2043 2027 5
2044 2028 5
2045 2029 5
2046 2030 4
2047 2031 4
2048 2032 4
2049 2033 4
2050 2034 4
2051 2035 5
2052 2036 5
2053 2037 5
2054 2038 4
2055 2039 4
2056 2040 4
2057 2041 4
2058 2042 4
2059 2043 4
2060 2044 4
2061 2045 4
2062 2046 4
2063 2047 4
2064 2048 4
2065 2049 4
2066 2050 4
2067 2051 4
2068 2052 4
2069 2053 4
2070 2054 4
2071 2055 5
2072 2056 5
2073 2057 5
2074 2058 5
2075 2059 5
2076 2060 5
2077 2061 5
2078 2062 5

FAO ('/OWNER=!%I', .OWNER_UIC);
FLAGS[B_PENDING_SPACES] = -1;

IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
AND .PROCESSOR_LENGTH NEQ 0
THEN
BEGIN
FAO ('/PROCESSOR=!AF',
.PROCESSOR_LENGTH, PROCESSOR);
FLAGS[B_PENDING_SPACES] = 1;
END;

FORMAT_PROTECTION(.PROTECTION);

IF .QUEUE_FLAGS[QUI$V_QUEUE_RETAIN_ALL]
THEN
BEGIN
FAO ('/RETAIN');
FLAGS[B_PENDING_SPACES] = 1;
END;

IF .QUEUE_FLAGS[QUI$V_QUEUE_RETAIN_ERROR]
THEN
BEGIN
FAO ('/RETAIN=ERROR');
FLAGS[B_PENDING_SPACES] = 1;
END;

FORMAT_SCHEDULE(.QUEUE_FLAGS);

IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
AND .ASSIGNED_QUEUE_NAME_LENGTH EQL 0
THEN
FORMAT_SEPARATE(.QUEUE_FLAGS,
.JOB_RESET_MODULES_LENGTH, JOB_RESET_MODULES);

IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
AND NOT .QUEUE_FLAGS[QUI$V_QUEUE_GENERIC]
THEN
BEGIN
IF .QUEUE_FLAGS[QUI$V_QUEUE_WSDEFAULT]
THEN
FORMAT_WORKING_SET(
UPCIT_BYTE-(%ASCIC 'WSDEFAULT'),
.WSDEFAULT);
```

```

EOL():
  FLAGS[B_PENDING_LINES] = 1;
  SHOW JOBS(.QUEUE_FLAGS, .QUEUE_STATUS);
  FLAGS[B_PENDING_LINES] = 1;
END;

```

														48	43	54	41	42	00D78	P.ACV:	.ASCII	\BATCH\	
																			00D7D		.BLKB	3	
																			00000005	P.ACU:	.LONG	5	
																			00000000		.ADDRESS	P.ACV	
														45	43	49	56	45	44	00D84	P.ACX:	.ASCII	\DEVICE\
																				00D8E	.BLKB	2	
																			00000006	P.ACW:	.LONG	6	
																			00000000		.ADDRESS	P.ACX	
68	63	74	61	62	20	63	69	72	65	6E	65	47	0D	00D98	P.ACY:	.ASCII	<13>\Generic batch\						
						6C	61	63	69	67	6F	4C	07	00DA6	P.ACZ:	.ASCII	<5>\Batch\						
72	65	76	72	65	73	20	63	69	72	65	6E	65	47	0E	00DAC	P.ADA:	.ASCII	<7>\Logical\					
6E	69	6D	72	65	74	20	63	69	72	65	6E	65	47	10	00DB4	P.ADB:	.ASCII	<14>\Generic server\					
												6C	61	00DC3	P.ADC:	.ASCII	<16>\Generic terminal\						
65	74	6E	69	72	70	20	63	69	72	65	6E	65	47	0F	00DD2								
													72	00DD4	P.ADD:	.ASCII	<15>\Generic printer\						
								72	65	76	72	65	53	06	00DE3								
61	6E	69	6D	72	65	74	20	65	74	6F	6D	65	52	0F	00DE4	P.ADE:	.ASCII	<6>\Server\					
															00DEB	P.ADF:	.ASCII	<15>\Remote terminal\					
						6C	61	6E	69	6D	72	65	54	08	00DFA								
						72	65	74	6E	69	72	50	07	00DFB	P.ADG:	.ASCII	<8>\Terminal\						
46	41	21	20	65	75	65	75	71	20	43	41	21	0D	00E04	P.ADH:	.ASCII	<7>\Printer\						
				67	6E	69	6E	67	69	6C	61	20	2C	0A	00E0C	P.ADI:	.ASCII	<13>\!AC queue !AF\					
72	65	73	20	72	6F	74	61	72	65	70	6F	20	2C	12	00E1A	P.ADJ:	.ASCII	<10>\, aligning\					
											65	63	69	76	00E25	P.ADK:	.ASCII	<18>\, operator service\					
						64	65	73	75	61	70	20	2C	08	00E34								
						67	6E	69	73	75	61	70	20	2C	09	00E38	P.ADL:	.ASCII	<8>\, paused\				
				67	6E	69	74	74	65	73	65	72	20	2C	0B	00E41	P.ADM:	.ASCII	<9>\, pausing\				
															00E4B	P.ADN:	.ASCII	<11>\, resetting\					

QUEMANSHO
V04-000

SHOW QUEUE command

M 5
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32:1

Page 64
(17)

67	6E	69	64	6E	65	70	20	70	6F	74	73	20	2C	0A	00E57	P.ADO:	.ASCII	<10>\, resuming\	
				67	64	65	70	70	6F	74	73	20	2C	09	00E62	P.ADP:	.ASCII	<9>\, stalled\	
					64	65	70	70	6F	74	73	20	2C	0A	00E6C	P.ADQ:	.ASCII	<10>\, starting\	
					67	6E	69	70	70	6F	74	73	20	2C	09	00E77	P.ADR:	.ASCII	<9>\, stopped\
61	76	61	6E	75	20	65	63	69	76	65	64	20	2C	0A	00E81	P.ADS:	.ASCII	<14>\, stop pending\	
															0A	00E90	P.ADT:	.ASCII	<10>\, stopping\
20	6F	74	20	64	65	6E	67	69	73	73	61	20	2C	14	00E9B	P.ADU:	.ASCII	<20>\, device unavailable\	
														69	00EAA				
														11	00EB0	P.ADV:	.ASCII	<17>\, assigned to !AF\	
														21	00EBF				
														05	00EC2	P.ADW:	.ASCII	<5>\, on \	
														05	00EC8	P.ADX:	.ASCII	<5>\!AF::\	
59	54	49	52	4F	49	52	50	5F	45	53	41	42	2F	12	00ECE	P.ADY:	.ASCII	<3>\!AF\	
														3D	00ED2	P.ADZ:	.ASCII	<18>\!BASE_PRIORITY=!UL\	
														16	00EE1				
28	3D	54	49	4D	49	4C	5F	4B	43	4F	4C	42	2F	16	00EE5	P.AEA:	.ASCII	<22>\!BLOCK_LIMIT=(!UL,!UL)\	
														21	00EF4				
28	3D	54	49	4D	49	4C	5F	4B	43	4F	4C	42	2F	15	00EFC	P.AEB:	.ASCII	<21>\!BLOCK_LIMIT=(!UL,"")\	
														21	00F0B				
21	3D	54	49	4D	49	4C	5F	4B	43	4F	4C	42	2F	10	00F12	P.AEC:	.ASCII	<16>\!BLOCK_LIMIT=!UL\	
														55	00F21				
														0A	00F23	P.AED:	.ASCII	<10>\CPUDEFAULT\	
														0A	00F2E	P.AEE:	.ASCII	<10>\CPUMAXIMUM\	
50	50	41	57	53	5F	45	4C	42	41	53	49	44	2F	11	00F39	P.AEF:	.ASCII	<17>\!DISABLE_SWAPPING\	
														49	00F48				
45	4E	45	47	5F	45	4C	42	41	4E	45	4F	4E	2F	11	00F4B	P.AEG:	.ASCII	<17>\!NOENABLE_GENERIC\	
														52	00F5A				
														09	00F5D	P.AEH:	.ASCII	<9>\!FORM=!AF\	
														0A	00F67	P.AEI:	.ASCII	<10>\!GENERIC=(\	
														04	00F72	P.AEJ:	.ASCII	<4>\!AF,\	
														03	00F77	P.AEK:	.ASCII	<3>\!AF\	
														01	00F7B	P.AEL:	.ASCII	<1>\!\	
4C	55	21	3D	54	49	4D	49	4C	5F	42	4F	4A	2F	0E	00F7D	P.AEM:	.ASCII	<14>\!JOB_LIMIT=!UL\	
														0C	00F8C	P.AEN:	.ASCII	<12>\!LIBRARY=!AF\	
														09	00F99	P.AEO:	.ASCII	<9>\Lowercase\	
														0A	00FA3	P.AEP:	.ASCII	<10>\!OWNER=!XI\	
46	41	21	3D	52	4F	53	53	45	43	4F	52	50	2F	0E	00FAE	P.AEQ:	.ASCII	<14>\!PROCESSOR=!AF\	
														07	00FBD	P.AER:	.ASCII	<7>\!RETAIN\	
														0D	00FC5	P.AES:	.ASCII	<13>\!RETAIN=ERROR\	
														09	00FD3	P.AET:	.ASCII	<9>\WSDEFAULT\	
														08	00FDD	P.AEU:	.ASCII	<8>\WSEXTENT\	
														07	00FE6	P.AEV:	.ASCII	<7>\WSQUOTA\	

OFFC 00000 SHOW_QUEUES:

5B	0000'	CF	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	1493
5A	0000V	CF	9E	00007	MOVAB	FLAGS+7, R11	
59	F8C1	CF	9E	0000C	MOVAB	FA0, R10	
5E	FB7C	CE	9E	00011	MOVAB	FORMAT WORKING_SET, R9	
		5B	D4	00016	CLRL	-1156(SP), SP	
20	AE	02	D0	00018	MOVL	QUEUE PRINTED	1562
		CF	9F	0001C	PUSHAB	#2, SEARCH_FLAGS	1565
00000000G	00	01	FB	00020	CALLS	P.ACU	1566
	04	50	E9	00027	BLBC	#1, CLISPRESNT	
20	AE	04	88	0002A	BISB2	R0, 1\$	
						#4, SEARCH_FLAGS	1567

00000000G	00	FD70	CF	9F	0002E	1\$:	PUSHAB	P.ACW	1568
	04		01	FB	00032		CALLS	#1, CLISPRESNT	
	20		50	E9	00039		BLBC	R0, 2\$	
FED0	CD	008D	08	88	0003C		BISB2	#8, SEARCH FLAGS	1569
FED2	CD	4D	CB	B0	00040	2\$:	MOVW	P2_DESC, ITMLST	1574
FED4	CD	0091	8F	9B	00047		MOVZBW	#77, ITMLST+2	1575
		FED8	CB	D0	0004D		MOVL	P2_DESC+4, ITMLST+4	1576
FEDC	CD	004C0004	CD	D4	00054		CLRL	ITMLST+8	1577
FEE0	CD	20	8F	D0	00058		MOVL	#4980740, ITMLST+12	1579
		FEE4	AE	9E	00061		MOVAB	SEARCH_FLAGS, ITMLST+16	1581
FEE8	CD	00030027	CD	D4	00067		CLRL	ITMLST+20	1582
FEED	CD	FEA8	8F	D0	0006B		MOVL	#196647, ITMLST+24	1584
FEF0	CD	FEA6	CD	9E	00074		MOVAB	ASSIGNED_QUEUE_NAME, ITMLST+28	1586
FEF4	CD	000E001F	CD	9E	0007B		MOVAB	ASSIGNED_QUEUE_NAME_LENGTH, ITMLST+32	1587
FEF8	CD	FE68	8F	D0	00082		MOVL	#917535, ITMLST+36	1589
FEFC	CD	FE66	CD	9E	0008B		MOVAB	DEVICE_NAME, ITMLST+40	1591
FF00	CD	00430027	CD	9E	00092		MOVAB	DEVICE_NAME_LENGTH, ITMLST+44	1592
FF04	CD	58	8F	D0	00099		MOVL	#4390951, ITMLST+48	1594
FF08	CD	56	AE	9E	000A2		MOVAB	PROCESSOR, ITMLST+52	1596
FF0C	CD	00450004	AE	9E	000A8		MOVAB	PROCESSOR_LENGTH, ITMLST+56	1597
FF10	CD	4C	8F	D0	000AE		MOVL	#4521988, ITMLST+60	1599
		FF14	AE	9E	000B7		MOVAB	QUEUE_FLAGS, ITMLST+64	1601
FF18	CD	0046001F	CD	D4	000BD		CLRL	ITMLST+68	1602
FF1C	CD	2C	8F	D0	000C1		MOVL	#4587551, ITMLST+72	1604
FF20	CD	2A	AE	9E	000CA		MOVAB	QUEUE_NAME, ITMLST+76	1606
FF24	CD	00470004	AE	9E	000D0		MOVAB	QUEUE_NAME_LENGTH, ITMLST+80	1607
FF28	CD	24	8F	D0	000D6		MOVL	#4653060, ITMLST+84	1609
		FF2C	AE	9E	000DF		MOVAB	QUEUE_STATUS, ITMLST+88	1611
FF30	CD	004B0006	CD	D4	000E5		CLRL	ITMLST+92	1612
FF34	CD	18	8F	D0	000E9		MOVL	#4915206, ITMLST+96	1614
FF38	CD	16	AE	9E	000F2		MOVAB	SCSNODE_NAME, ITMLST+100	1616
		FF3C	AE	9E	000F8		MOVAB	SCSNODE_NAME_LENGTH, ITMLST+104	1617
03	F9	AB	CD	D4	000FE		CLRL	ITMLST+108	1619
			01	E0	00102		BBS	#1, FLAGS, 3\$	1622
			0129	31	00107		BRW	4\$	
FF3C	CD	00040004	8F	D0	0010A	3\$:	MOVL	#262148, ITMLST+108	1625
FF40	CD	FEA0	CD	9E	00113		MOVAB	BASE_PRIORITY, ITMLST+112	1627
		FF44	CD	D4	0011A		CLRL	ITMLST+116	1628
FF48	CD	00070010	8F	D0	0011E		MOVL	#458768, ITMLST+120	1630
FF4C	CD	FE90	CD	9E	00127		MOVAB	CHARACTERISTICS, ITMLST+124	1632
		FF50	CD	D4	0012E		CLRL	ITMLST+128	1633
FF54	CD	000C0004	8F	D0	00132		MOVL	#786436, ITMLST+132	1635
FF58	CD	FE8C	CD	9E	0013B		MOVAB	CPU_DEFAULT, ITMLST+136	1637
		FF5C	CD	D4	00142		CLRL	ITMLST+140	1638
FF60	CD	000D0004	8F	D0	00146		MOVL	#851972, ITMLST+144	1640
FF64	CD	FE88	CD	9E	0014F		MOVAB	CPU_LIMIT, ITMLST+148	1642
		FF68	CD	D4	00156		CLRL	ITMLST+152	1643
FF6C	CD	001F001F	8F	D0	0015A		MOVL	#2031647, ITMLST+156	1645
FF70	CD	FE44	CD	9E	00163		MOVAB	FORM_NAME, ITMLST+160	1647
FF74	CD	FE42	CD	9E	0016A		MOVAB	FORM_NAME_LENGTH, ITMLST+164	1648
FF78	CD	002400FF	8F	D0	00171		MOVL	#2359551, ITMLST+168	1650
FF7C	CD	01C4	CE	9E	0017A		MOVAB	GENERIC_TARGET, ITMLST+172	1652
80	AD	01C2	CE	9E	00181		MOVAB	GENERIC_TARGET_LENGTH, ITMLST+176	1653
84	AD	002B0004	8F	D0	00187		MOVL	#2818052, ITMLST+180	1655
88	AD	01BC	CE	9E	0018F		MOVAB	JOB_LIMIT, ITMLST+184	1657
		8C	AD	D4	00195		CLRL	ITMLST+188	1658
90	AD	002D00FF	8F	D0	00198		MOVL	#2949375, ITMLST+192	1660

94	AD	00BC	CE	9E	001A0	MOVAB	JOB-RESET_MODULES, ITMLST+196	:	1662
98	AD	00BA	CE	9E	001A6	MOVAB	JOB-RESET_MODULES_LENGTH, ITMLST+200	:	1663
9C	AD	002F0004	8F	D0	001AC	MOVL	#3080196, ITMLST+204	:	1665
A0	AD	00B4	CE	9E	001B4	MOVAB	JOB_SIZE_MAXIMUM, ITMLST+208	:	1667
		A4	AD	D4	001BA	CLRL	ITMLST+212	:	1668
A8	AD	00300004	8F	D0	001BD	MOVL	#3145732, ITMLST+216	:	1670
AC	AD	00B0	CE	9E	001C5	MOVAB	JOB_SIZE_MINIMUM, ITMLST+220	:	1672
		B0	AD	D4	001CB	CLRL	ITMLST+224	:	1673
B4	AD	00330027	8F	D0	001CE	MOVL	#3342375, ITMLST+228	:	1675
B8	AD	0088	CE	9E	001D6	MOVAB	LIBRARY_SPECIFICATION, ITMLST+232	:	1677
BC	AD	0086	CE	9E	001DC	MOVAB	LIBRARY_SPECIFICATION_LENGTH, ITMLST+236	:	1678
C0	AD	00380004	8F	D0	001E2	MOVL	#3670020, ITMLST+240	:	1680
C4	AD	0080	CE	9E	001EA	MOVAB	OWNER_UID, ITMLST+244	:	1682
		C8	AD	D4	001F0	CLRL	ITMLST+248	:	1683
CC	AD	00440004	8F	D0	001F3	MOVL	#4456452, ITMLST+252	:	1685
D0	AD	50	AE	9E	001FB	MOVAB	PROTECTION, ITMLST+256	:	1687
		D4	AD	D4	00200	CLRL	ITMLST+260	:	1688
D8	AD	00520004	8F	D0	00203	MOVL	#5373956, ITMLST+264	:	1690
DC	AD	10	AE	9E	0020B	MOVAB	WSDEFAULT, ITMLST+268	:	1692
		E0	AD	D4	00210	CLRL	ITMLST+272	:	1693
E4	AD	00530004	8F	D0	00213	MOVL	#5439492, ITMLST+276	:	1695
E8	AD	0C	AE	9E	0021B	MOVAB	WSEXTENT, ITMLST+280	:	1697
		EC	AD	D4	00220	CLRL	ITMLST+284	:	1698
F0	AD	00540004	8F	D0	00223	MOVL	#5505028, ITMLST+288	:	1700
F4	AD	08	AE	9E	0022B	MOVAB	WSQUOTA, ITMLST+292	:	1702
		F8	AD	7C	00230	CLRL	ITMLST+296	:	1703
			7E	7C	00233	CLRL	-(SP)	:	1716
		08	AE	9F	00235	PUSHAB	IOSB	:	
		FED0	CD	9F	00238	PUSHAB	ITMLST	:	
		7E	06	7D	0023C	MOVQ	#6, -(SP)	:	
			7E	D4	0023F	CLRL	-(SP)	:	
00000000G	00		07	FB	00241	CALLS	#7, SYSSGETQUIW	:	
	57		50	D0	00248	MOVL	R0, STATUS	:	
	03		57	E9	0024B	BLBC	STATUS, 5\$:	1717
	57		6E	D0	0024E	MOVL	IOSB, STATUS	:	
	50		57	D2	00251	MCOML	STATUS, R0	:	1723
0D	58		00	E2	00254	BBSS	#0, QUEUE_PRINTED, 6\$:	1720
	0E		50	E9	00258	BLBC	R0, 7\$:	1723
			57	DD	0025B	PUSHL	STATUS	:	1726
00000000G	00		01	FB	0025D	CALLS	#1, LIBSSIGNAL	:	
				04	00264	RET		:	1725
	01		50	E9	00265	BLBC	R0, 7\$:	1732
				04	00268	RET		:	
07	13	4C	AE	E9	00269	BLBC	QUEUE_FLAGS, 9\$:	1736
	AE		01	E1	0026D	BBBC	#1, QUEUE_FLAGS+1, 8\$:	1738
	56	FB34	CF	9E	00272	MOVAB	P.ACY, TYPE	:	1740
			61	11	00277	BRB	17\$:	
	56	FB3B	CF	9E	00279	MOVAB	P.ACZ, TYPE	:	1742
			5A	11	0027E	BRB	17\$:	1738
		FEA6	CD	B5	00280	TSTW	ASSIGNED_QUEUE_NAME_LENGTH	:	1744
			07	13	00284	BEQL	10\$:	
	56	FB34	CF	9E	00286	MOVAB	P.ADA, TYPE	:	1746
			4D	11	0028B	BRB	17\$:	
1F	4D	AE	01	E1	0028D	BBBC	#1, QUEUE_FLAGS+1, 13\$:	1748
		56	AE	B5	00292	TSTW	PROCESSOR_LENGTH	:	1750
			07	13	00295	BEQL	11\$:	
	56	FB2B	CF	9E	00297	MOVAB	P.ADB, TYPE	:	1752

07	4E	AE	3C	11	0029C	BRB	17\$		
		56	02	E1	0029E	BBC	#2, QUEUE_FLAGS+2, 12\$	1753	
			CF	9E	002A3	MOVAB	P.ADC, TYPE	1755	
			30	11	002A8	BRB	17\$		
		56	CF	9E	002AA	MOVAB	P.ADD, TYPE	1757	
			29	11	002AF	BRB	17\$	1750	
07	25	AE	01	E1	002B1	BBC	#1, QUEUE_STATUS+1, 14\$	1759	
		56	CF	9E	002B6	MOVAB	P.ADE, TYPE	1761	
			1D	11	002BB	BRB	17\$		
13	4E	AE	02	E1	002BD	BBC	#2, QUEUE_FLAGS+2, 16\$	1762	
07	24	AE	06	E1	002C2	BBC	#6, QUEUE_STATUS, 15\$	1764	
		56	CF	9E	002C7	MOVAB	P.ADF, TYPE	1766	
			0C	11	002CC	BRB	17\$		
		56	CF	9E	002CE	MOVAB	P.ADG, TYPE	1768	
			05	11	002D3	BRB	17\$	1764	
		56	CF	9E	002D5	MOVAB	P.ADH, TYPE	1770	
	F9	AB	04	88	002DA	BISB2	#4, FLAGS	1773	
		2C	AE	9F	002DE	PUSHAB	QUEUE_NAME	1774	
		7E	AE	3C	002E1	MOVZWL	QUEUE_NAME_LENGTH, -(SP)		
			56	DD	002E5	PUSHL	TYPE		
			CF	9F	002E7	PUSHAB	P.ADI		
		6A	04	FB	002EB	CALLS	#4, FAO		
		07	AE	E9	002EE	BLBC	QUEUE_STATUS, 18\$	1777	
			CF	9F	002F2	PUSHAB	P.ADJ	1778	
		6A	01	FB	002F6	CALLS	#1, FAO		
07	24	AE	03	E1	002F9	BBC	#3, QUEUE_STATUS, 19\$	1779	
			CF	9F	002FE	PUSHAB	P.ADK	1780	
		6A	01	FB	00302	CALLS	#1, FAO		
07	24	AE	04	E1	00305	BBC	#4, QUEUE_STATUS, 20\$	1781	
			CF	9F	0030A	PUSHAB	P.ADL	1782	
		6A	01	FB	0030E	CALLS	#1, FAO		
07	24	AE	05	E1	00311	BBC	#5, QUEUE_STATUS, 21\$	1783	
			CF	9F	00316	PUSHAB	P.ADM	1784	
		6A	01	FB	0031A	CALLS	#1, FAO		
		24	AE	95	0031D	TSTB	QUEUE_STATUS	1785	
			07	18	00320	BGEQ	22\$		
			CF	9F	00322	PUSHAB	P.ADN	1786	
		6A	01	FB	00326	CALLS	#1, FAO		
		0B	AE	E8	00329	BLBS	QUEUE_STATUS, 23\$	1787	
		07	AE	E9	0032D	BLBC	QUEUE_STATUS+1, 23\$	1788	
			CF	9F	00331	PUSHAB	P.ADO	1789	
		6A	01	FB	00335	CALLS	#1, FAO		
07	25	AE	02	E1	00338	BBC	#2, QUEUE_STATUS+1, 24\$	1790	
			CF	9F	0033D	PUSHAB	P.ADP	1791	
		6A	01	FB	00341	CALLS	#1, FAO		
07	25	AE	03	E1	00344	BBC	#3, QUEUE_STATUS+1, 25\$	1792	
			CF	9F	00349	PUSHAB	P.ADQ	1793	
		6A	01	FB	0034D	CALLS	#1, FAO		
12	25	AE	04	E1	00350	BBC	#4, QUEUE_STATUS+1, 28\$	1794	
06	24	AE	01	E1	00355	BBC	#1, QUEUE_STATUS, 26\$	1796	
			CF	9F	0035A	PUSHAB	P.ADR	1797	
			04	11	0035E	BRB	27\$		
			CF	9F	00360	PUSHAB	P.ADS	1798	
		6A	01	FB	00364	CALLS	#1, FAO		
07	25	AE	05	E1	00367	BBC	#5, QUEUE_STATUS+1, 29\$	1799	
			CF	9F	0036C	PUSHAB	P.ADT	1800	
		6A	01	FB	00370	CALLS	#1, FAO		

07	25	AE		06	E1	00373	29\$:	BBC	#6, QUEUE_STATUS+1, 30\$	1801
		6A	FB31	CF	9F	00378		PUSHAB	P.ADU	1802
				01	FB	0037C		CALLS	#1, FAO	
			FEA6	CD	B5	0037F	30\$:	TSTW	ASSIGNED_QUEUE_NAME_LENGTH	1805
				10	13	00383		BEQL	31\$	
			FEA8	CD	9F	00385		PUSHAB	ASSIGNED_QUEUE_NAME	1808
		7E	FEA6	CD	3C	00389		MOVZWL	ASSIGNED_QUEUE_NAME_LENGTH, -(SP)	
			FB30	CF	9F	0038E		PUSHAB	P.ADV	
		6A		03	FB	00392		CALLS	#3, FAO	
			FEA6	CD	B5	00395	31\$:	TSTW	ASSIGNED_QUEUE_NAME_LENGTH	1811
				40	12	00399		BNEQ	35\$	
3B	4D	AE		01	E0	0039B		BBS	#1, QUEUE_FLAGS+1, 35\$	1812
			16	AE	B5	003A0		TSTW	SCSNODE_NAME_LENGTH	1815
				06	12	003A3		BNEQ	32\$	
			FE66	CD	B5	003A5		TSTW	DEVICE_NAME_LENGTH	
				07	13	003A9		BEQL	33\$	
			FB25	CF	9F	003AB	32\$:	PUSHAB	P.ADW	1817
		6A		01	FB	003AF		CALLS	#1, FAO	
			16	AE	B5	003B2	33\$:	TSTW	SCSNODE_NAME_LENGTH	1820
				0E	13	003B5		BEQL	34\$	
			18	AE	9F	003B7		PUSHAB	SCSNODE_NAME	1822
		7E	1A	AE	3C	003BA		MOVZWL	SCSNODE_NAME_LENGTH, -(SP)	
			FB18	CF	9F	003BE		PUSHAB	P.ADX	
		6A		03	FB	003C2		CALLS	#3, FAO	
			FE66	CD	B5	003C5	34\$:	TSTW	DEVICE_NAME_LENGTH	1825
				10	13	003C9		BEQL	35\$	
			FE68	CD	9F	003CB		PUSHAB	DEVICE_NAME	1827
		7E	FE66	CD	3C	003CF		MOVZWL	DEVICE_NAME_LENGTH, -(SP)	
			FB08	CF	9F	003D4		PUSHAB	P.ADY	
		6A		03	FB	003D8		CALLS	#3, FAO	
03	F9	AB		04	8A	003DB	35\$:	BICB2	#4, FLAGS	1829
	F9	AB		01	E0	003DF		BBS	#1, FLAGS, 36\$	1832
				0278	31	003E4		BRW	64\$	
	0000V	CF		00	FB	003E7	36\$:	CALLS	#0, EOL	1835
		6B		04	90	003EC		MOVB	#4, FLAGS+7	1836
	FD	AB		04	90	003EF		MOVB	#4, FLAGS+4	1837
14	4D	AE		01	E0	003F3		BBS	#1, QUEUE_FLAGS+1, 37\$	1840
			FEA6	CD	B5	003F8		TSTW	ASSIGNED_QUEUE_NAME_LENGTH	1841
				0E	12	003FC		BNEQ	37\$	
			FEA0	CD	DD	003FE		PUSHL	BASE_PRIORITY	1844
			FADE	CF	9F	00402		PUSHAB	P.ADZ	
		6A		02	FB	00406		CALLS	#2, FAO	
		6B		01	90	00409		MOVB	#1, FLAGS+7	1845
		46	4C	AE	E8	0040C	37\$:	BLBS	QUEUE_FLAGS, 42\$	1849
41	4D	AE		01	E0	00410		BBS	#1, QUEUE_FLAGS+1, 42\$	1850
			FEA6	CD	B5	00415		TSTW	ASSIGNED_QUEUE_NAME_LENGTH	1851
				3B	12	00419		BNEQ	42\$	
			00B0	CE	D5	0041B		TSTL	JOB_SIZE_MINIMUM	1854
				21	13	0041F		BEQL	39\$	
			00B4	CE	D5	00421		TSTL	JOB_SIZE_MAXIMUM	1856
				11	13	00425		BEQL	38\$	
			00B4	CE	DD	00427		PUSHL	JOB_SIZE_MAXIMUM	1860
			00B4	CE	DD	0042B		PUSHL	JOB_SIZE_MINIMUM	
			FAC4	CF	9F	0042F		PUSHAB	P.AEA	
		6A		03	FB	00433		CALLS	#3, FAO	
				1B	11	00436		BRB	41\$	1856
			00B0	CE	DD	00438	38\$:	PUSHL	JOB_SIZE_MINIMUM	1866

			FACE	CF	9F	0043C		PUSHAB	P.AEB		
				0E	11	00440		BRB	40\$		
			00B4	CE	D5	00442	39\$:	TSTL	JOB_SIZE_MAXIMUM		1870
				0E	13	00446		BEQL	42\$		
			00B4	CE	DD	00448		PUSHL	JOB_SIZE_MAXIMUM		1874
			FAD4	CF	9F	0044C		PUSHAB	P.AEC		
		6A		02	FB	00450	40\$:	CALLS	#2, FAO		
		6B		01	90	00453	41\$:	MOVB	#1, FLAGS+7		1875
OF	4D	AE		01	E0	00456	42\$:	BBS	#1, QUEUE_FLAGS+1, 43\$		1880
			FEA6	CD	B5	0045B		TSTW	ASSIGNED_QUEUE_NAME_LENGTH		1881
				09	12	0045F		BNEQ	43\$		
			FE90	CD	9F	00461		PUSHAB	CHARACTERISTICS		1883
	FA48	C9		01	FB	00465		CALLS	#1, FORMAT_CHARACTERISTICS		
		29	4C	AE	E9	0046A	43\$:	BLBC	QUEUE_FLAGS, 45\$		1886
24	4D	AE		01	E0	0046E		BBS	#1, QUEUE_FLAGS+1, 45\$		1887
OD	4C	AE		01	E1	00473		BBC	#1, QUEUE_FLAGS, 44\$		1890
			FE8C	CD	DD	00478		PUSHL	CPU_DEFAULT		1894
			FAB5	CF	9F	0047C		PUSHAB	P.AED		1893
	FAB8	C9		02	FB	00480		CALLS	#2, FORMAT_CPU_TIME		
OD	4C	AE		02	E1	00485	44\$:	BBC	#2, QUEUE_FLAGS, 45\$		1897
			FE88	CD	DD	0048A		PUSHL	CPU_LIMIT		1901
	FAB8	C9		CF	9F	0048E		PUSHAB	P.AEE		1900
		13	4C	02	FB	00492		CALLS	#2, FORMAT_CPU_TIME		
OE	4D	AE		AE	E8	00497	45\$:	BLBS	QUEUE_FLAGS, 46\$		1905
				01	E0	0049B		BBS	#1, QUEUE_FLAGS+1, 46\$		1906
			FEA6	CD	B5	004A0		TSTW	ASSIGNED_QUEUE_NAME_LENGTH		1907
				08	12	004A4		BNEQ	46\$		
			4C	AE	DD	004A6		PUSHL	QUEUE_FLAGS		1909
	FBA0	C9		01	FB	004A9		CALLS	#1, FORMAT_DEFAULT		
		14	4C	AE	E9	004AE	46\$:	BLBC	QUEUE_FLAGS, 47\$		1912
OF	4D	AE		01	E0	004B2		BBS	#1, QUEUE_FLAGS+1, 47\$		1913
OA	4E	AE		01	E0	004B7		BBS	#1, QUEUE_FLAGS+2, 47\$		1916
			FA8B	CF	9F	004BC		PUSHAB	P.AEF		1919
		6A		01	FB	004C0		CALLS	#1, FAO		
		6B		01	90	004C3		MOVB	#1, FLAGS+7		1920
15	4D	AE		01	E0	004C6	47\$:	BBS	#1, QUEUE_FLAGS+1, 48\$		1925
			FEA6	CD	B5	004CB		TSTW	ASSIGNED_QUEUE_NAME_LENGTH		1926
				0F	12	004CF		BNEQ	48\$		
OA	4D	AE		02	E0	004D1		BBS	#2, QUEUE_FLAGS+1, 48\$		1929
			FA83	CF	9F	004D6		PUSHAB	P.AEG		1932
		6A		01	FB	004DA		CALLS	#1, FAO		
		6B		01	90	004DD		MOVB	#1, FLAGS+7		1933
		1E	4C	AE	E8	004E0	48\$:	BLBS	QUEUE_FLAGS, 49\$		1938
19	4D	AE		01	E0	004E4		BBS	#1, QUEUE_FLAGS+1, 49\$		1939
			FEA6	CD	B5	004E9		TSTW	ASSIGNED_QUEUE_NAME_LENGTH		1940
				13	12	004ED		BNEQ	49\$		
			FE44	CD	9F	004EF		PUSHAB	FORM_NAME		1943
		7E	FE42	CD	3C	004F3		MOVZWL	FORM_NAME_LENGTH, -(SP)		
			FA73	CF	9F	004F8		PUSHAB	P.AEH		
		6A		03	FB	004FC		CALLS	#3, FAO		
		6B		01	90	004FF		MOVB	#1, FLAGS+7		1944
5A	4D	AE		01	E1	00502	49\$:	BBC	#1, QUEUE_FLAGS+1, 53\$		1948
			01C2	CE	B5	00507		TSTW	GENERIC_TARGET_LENGTH		1949
				54	13	0050B		BEQL	53\$		
			FA68	CF	9F	0050D		PUSHAB	P.AEI		1958
		6A		01	FB	00511		CALLS	#1, FAO		
				6B	94	00514		CLRB	FLAGS+7		1959

		53	01C4	CE	9E	00516	MOVAB	GENERIC TARGET, REMAINING ADDRESS	1961
		54		53	DO	0051B	MOVL	REMAINING ADDRESS, ITEM ADDRESS	
63		52	01C2	CE	3C	0051E	MOVZWL	GENERIC TARGET_LENGTH, REMAINING_LENGTH	1962
		52		2C	3A	00523	LOCC	#44, REMAINING_LENGTH, (REMAINING_ADDRESS)	1964
				02	12	00527	BNEQ	51\$	
				51	D4	00529	CLRL	R1	
		53		51	DO	0052B	MOVL	R1, REMAINING_ADDRESS	
55		53		1E	13	0052E	BEQL	52\$	1965
				54	C3	00530	SUBL3	ITEM_ADDRESS, REMAINING_ADDRESS, -	1967
								ITEM_LENGTH	
				54	DD	00534	PUSHL	ITEM_ADDRESS	1968
				55	DD	00536	PUSHL	ITEM_LENGTH	
			FA48	CF	9F	00538	PUSHAB	P.AEJ	
		6A		03	FB	0053C	CALLS	#3, FAO	
50		52		55	C3	0053F	SUBL3	ITEM_LENGTH, REMAINING_LENGTH, R0	1969
		52	FF	A0	9E	00543	MOVAB	-1(R0), REMAINING_LENGTH	
				53	D6	00547	INCL	REMAINING_ADDRESS	1970
		54		53	DO	00549	MOVL	REMAINING_ADDRESS, ITEM_ADDRESS	
				D5	11	0054C	BRB	50\$	1964
				14	BB	0054E	PUSHR	#*M<R2,R4>	1973
			FA35	CF	9F	00550	PUSHAB	P.AEK	
		6A		03	FB	00554	CALLS	#3, FAO	
			FA32	CF	9F	00557	PUSHAB	P.AEL	1974
		6A		01	FB	0055B	CALLS	#1, FAO	
		6B		01	90	0055E	MOVB	#1, FLAGS+7	1975
		13	4C	AE	E9	00561	BLBC	QUEUE_FLAGS, 54\$	1979
OE	4D	AE		01	E0	00565	BBS	#1, QUEUE_FLAGS+1, 54\$	1980
			01BC	CE	DD	0056A	PUSHL	JOB LIMIT	1983
			FA1D	CF	9F	0056E	PUSHAB	P.AEM	
		6A		02	FB	00572	CALLS	#2, FAO	
		6B		01	90	00575	MOVB	#1, FLAGS+7	1984
		24	4C	AE	E8	00578	BLBS	QUEUE_FLAGS, 55\$	1988
1F	4D	AE		01	E0	0057C	BBS	#1, QUEUE_FLAGS+1, 55\$	1989
			FEA6	CD	B5	00581	TSTW	ASSIGNED_QUEUE_NAME_LENGTH	1990
				19	12	00585	BNEQ	55\$	
			0086	CE	B5	00587	TSTW	LIBRARY_SPECIFICATION_LENGTH	1991
				13	13	0058B	BEQL	55\$	
			0088	CE	9F	0058D	PUSHAB	LIBRARY_SPECIFICATION	1995
		7E	008A	CE	3C	00591	MOVZWL	LIBRARY_SPECIFICATION_LENGTH, -(SP)	
			FA04	CF	9F	00596	PUSHAB	P.AEN	
		6A		03	FB	0059A	CALLS	#3, FAO	
		6B		01	90	0059D	MOVB	#1, FLAGS+7	1996
OA	24	AE		02	E1	005A0	BBC	#2, QUEUE_STATUS, 56\$	2000
			FA02	CF	9F	005A5	PUSHAB	P.AEO	2003
		6A		01	FB	005A9	CALLS	#1, FAO	
		6B		01	90	005AC	MOVB	#1, FLAGS+7	2004
			0080	CE	DD	005AF	PUSHL	OWNER_UIC	2008
			F9FE	CF	9F	005B3	PUSHAB	P.AEP	
		6A		02	FB	005B7	CALLS	#2, FAO	
		6B		01	90	005BA	MOVB	#1, FLAGS+7	2009
		16	4C	AE	E8	005BD	BLBS	QUEUE_FLAGS, 57\$	2012
			56	AE	B5	005C1	TSTW	PROCESSOR_LENGTH	2013
				11	13	005C4	BEQL	57\$	
			58	AE	9F	005C6	PUSHAB	PROCESSOR	2017
		7E	5A	AE	3C	005C9	MOVZWL	PROCESSOR_LENGTH, -(SP)	
			F9EF	CF	9F	005CD	PUSHAB	P.AEQ	
		6A		03	FB	005D1	CALLS	#3, FAO	

		6B		01	90	005D4		MOVB	#1, FLAGS+7		2018
			50	AE	DD	005D7	57\$:	PUSHL	PROTECTION		2022
	FE20	C9		01	FB	005DA		CALLS	#1, FORMAT_PROTECTION		
			4D	AE	95	005DF		TSTB	QUEUE_FLAGS+1		2025
				0A	18	005E2		BGEQ	58\$		
			F9E7	CF	9F	005E4		PUSHAB	P.AER		2028
		6A		01	FB	005E8		CALLS	#1, FAO		
		6B		01	90	005EB		MOVB	#1, FLAGS+7		2029
		0A	4E	AE	E9	005EE	58\$:	BLBC	QUEUE_FLAGS+2, 59\$		2033
			F9E1	CF	9F	005F2		PUSHAB	P.AES		2036
		6A		01	FB	005F6		CALLS	#1, FAO		
		6B		01	90	005F9		MOVB	#1, FLAGS+7		2037
			4C	AE	DD	005FC	59\$:	PUSHL	QUEUE_FLAGS		2041
	FEB9	C9		01	FB	005FF		CALLS	#1, FORMAT_SCHEDULE		
		1C	4C	AE	E8	00604		BLBS	QUEUE_FLAGS, 60\$		2044
17	4D	AE		01	E0	00608		BBS	#1, QUEUE_FLAGS+1, 60\$		2045
			FEA6	CD	B5	0060D		TSTW	ASSIGNED_QUEUE_NAME_LENGTH		2046
				11	12	00611		BNEQ	60\$		
			00BC	CE	9F	00613		PUSHAB	JOB_RESET_MODULES		2048
		7E	00BE	CE	3C	00617		MOVZWL	JOB_RESET_MODULES_LENGTH, -(SP)		2049
			54	AE	DD	0061C		PUSHL	QUEUE_FLAGS		2048
	FF42	C9		03	FB	0061F		CALLS	#3, FORMAT_SEPARATE		
		32	4C	AE	E9	00624	60\$:	BLBC	QUEUE_FLAGS, 63\$		2052
2D	4D	AE		01	E0	00628		BBS	#1, QUEUE_FLAGS+1, 63\$		2053
0A	4E	AE		03	E1	0062D		BBC	#3, QUEUE_FLAGS+2, 61\$		2056
			10	AE	DD	00632		PUSHL	WSDEFAULT		2060
			F9AC	CF	9F	00635		PUSHAB	P.AET		2059
		69		02	FB	00639		CALLS	#2, FORMAT_WORKING_SET		
0A	4E	AE		04	E1	0063C	61\$:	BBC	#4, QUEUE_FLAGS+2, 62\$		2063
			0C	AE	DD	00641		PUSHL	WSXTENT		2067
			F9A7	CF	9F	00644		PUSHAB	P.AEU		2066
		69		02	FB	00648		CALLS	#2, FORMAT_WORKING_SET		
0A	4E	AE		05	E1	0064B	62\$:	BBC	#5, QUEUE_FLAGS+2, 63\$		2070
			08	AE	DD	00650		PUSHL	WSQUOTA		2074
			F9A1	CF	9F	00653		PUSHAB	P.AEV		2073
		69		02	FB	00657		CALLS	#2, FORMAT_WORKING_SET		
				6B	94	0065A	63\$:	CLRB	FLAGS+7		2078
			FD	AB	94	0065C		CLRB	FLAGS+4		2079
	0000V	CF		00	FB	0065F	64\$:	CALLS	#0, EOL		2083
	FF	AB		01	90	00664		MOVB	#1, FLAGS+6		2084
			24	AE	DD	00668		PUSHL	QUEUE_STATUS		2085
			50	AE	DD	0066B		PUSHL	QUEUE_FLAGS		
	0000V	CF		02	FB	0066E		CALLS	#2, SHOW_JOBS		
	FF	AB		01	90	00673		MOVB	#1, FLAGS+6		2086
			FBB9	31	00677			BRW	4\$		1711
				04	0067A			RET			2088

; Routine Size: 1659 bytes, Routine Base: CODE + 0FEE


```
2106 2089 1 ROUTINE SHOW_JOBS(QUEUE_FLAGS,QUEUE_STATUS): NOVALUE=
2107 2090 1
2108 2091 1 ++
2109 2092 1
2110 2093 1 FUNCTIONAL DESCRIPTION:
2111 2094 1 This routine displays the jobs in a queue.
2112 2095 1
2113 2096 1 INPUT PARAMETERS:
2114 2097 1 NONE
2115 2098 1
2116 2099 1 IMPLICIT INPUTS:
2117 2100 1 NONE
2118 2101 1
2119 2102 1 OUTPUT PARAMETERS:
2120 2103 1 NONE
2121 2104 1
2122 2105 1 IMPLICIT OUTPUTS:
2123 2106 1 NONE
2124 2107 1
2125 2108 1 ROUTINE VALUE:
2126 2109 1 NONE
2127 2110 1
2128 2111 1 SIDE EFFECTS:
2129 2112 1 NONE
2130 2113 1
2131 2114 1 --
2132 2115 1
2133 2116 2 BEGIN
2134 2117 2 MAP
2135 2118 2 QUEUE_FLAGS: BBLOCK,
2136 2119 2 QUEUE_STATUS: BBLOCK;
2137 2120 2
2138 2121 2 LOCAL
2139 2122 2 ITMLST: BBLOCK[448],
2140 2123 2 AFTER TIME: VOLATILE VECTOR[2],
2141 2124 2 CHARACTERISTICS: VOLATILE BITVECTOR[128],
2142 2125 2 CHECKPOINT_DATA: VOLATILE VECTOR[255,BYTE],
2143 2126 2 CHECKPOINT_DATA_LENGTH: VOLATILE WORD,
2144 2127 2 CLI: VOLATILE VECTOR[39,BYTE],
2145 2128 2 CLI_LENGTH: VOLATILE WORD,
2146 2129 2 COMPLETED_BLOCKS: VOLATILE,
2147 2130 2 CONDITION_VECTOR: VOLATILE VECTOR[3],
2148 2131 2 CPU_LIMIT: VOLATILE,
2149 2132 2 ENTRY_NUMBER: VOLATILE,
2150 2133 2 FORM_NAME: VOLATILE VECTOR[31,BYTE],
2151 2134 2 FORM_NAME_LENGTH: VOLATILE WORD,
2152 2135 2 INTERVENING_BLOCKS: VOLATILE,
2153 2136 2 INTERVENING_JOBS: VOLATILE,
2154 2137 2 JOB_COPIES: VOLATILE,
2155 2138 2 JOB_COPIES_DONE: VOLATILE,
2156 2139 2 JOB_FLAGS: VOLATILE BBLOCK[4],
2157 2140 2 JOB_NAME: VOLATILE VECTOR[39,BYTE],
2158 2141 2 JOB_NAME_LENGTH: VOLATILE WORD,
2159 2142 2 JOB_SIZE: VOLATILE,
2160 2143 2 JOB_STATUS: VOLATILE BBLOCK[4],
2161 2144 2 LOG_QUEUE: VOLATILE VECTOR[39,BYTE],
2162 2145 2 LOG_QUEUE_LENGTH: VOLATILE WORD,
LOG_SPECIFICATION: VOLATILE VECTOR[255,BYTE],
```

```
: 2163      2146 2      LOG SPECIFICATION_LENGTH:      VOLATILE WORD,
: 2164      2147 2      NOTE:                        VOLATILE VECTOR[255,BYTE],
: 2165      2148 2      NOTE_LENGTH:                  VOLATILE WORD,
: 2166      2149 2      OPERATOR_REQUEST:              VOLATILE VECTOR[255,BYTE],
: 2167      2150 2      OPERATOR_REQUEST_LENGTH:        VOLATILE WORD,
: 2168      2151 2      PARAMETER_1:                   VOLATILE VECTOR[255,BYTE],
: 2169      2152 2      PARAMETER_1_LENGTH:             VOLATILE WORD,
: 2170      2153 2      PARAMETER_2:                   VOLATILE VECTOR[255,BYTE],
: 2171      2154 2      PARAMETER_2_LENGTH:             VOLATILE WORD,
: 2172      2155 2      PARAMETER_3:                   VOLATILE VECTOR[255,BYTE],
: 2173      2156 2      PARAMETER_3_LENGTH:             VOLATILE WORD,
: 2174      2157 2      PARAMETER_4:                   VOLATILE VECTOR[255,BYTE],
: 2175      2158 2      PARAMETER_4_LENGTH:             VOLATILE WORD,
: 2176      2159 2      PARAMETER_5:                   VOLATILE VECTOR[255,BYTE],
: 2177      2160 2      PARAMETER_5_LENGTH:             VOLATILE WORD,
: 2178      2161 2      PARAMETER_6:                   VOLATILE VECTOR[255,BYTE],
: 2179      2162 2      PARAMETER_6_LENGTH:             VOLATILE WORD,
: 2180      2163 2      PARAMETER_7:                   VOLATILE VECTOR[255,BYTE],
: 2181      2164 2      PARAMETER_7_LENGTH:             VOLATILE WORD,
: 2182      2165 2      PARAMETER_8:                   VOLATILE VECTOR[255,BYTE],
: 2183      2166 2      PARAMETER_8_LENGTH:             VOLATILE WORD,
: 2184      2167 2      PRIORITY:                      VOLATILE,
: 2185      2168 2      REFUSAL_REASON:                 VOLATILE VECTOR[255,BYTE],
: 2186      2169 2      REFUSAL_REASON_LENGTH:          VOLATILE WORD,
: 2187      2170 2      SEARCH_FLAGS:                  VOLATILE BBLOCK[4],
: 2188      2171 2      SUBMISSION_TIME:               VOLATILE VECTOR[2],
: 2189      2172 2      USERNAME:                     VOLATILE VECTOR[12,BYTE],
: 2190      2173 2      USERNAME_LENGTH:               VOLATILE WORD,
: 2191      2174 2      WSDEFAULT:                    VOLATILE,
: 2192      2175 2      WSEXTENT:                     VOLATILE,
: 2193      2176 2      WSQUOTA:                      VOLATILE,
: 2194      2177 2      STATUS,
: 2195      2178 2      IOSB:                          VECTOR[2],
: 2196      2179 2      JOB_PRINTED:
: 2197      2180
: 2198      2181
: 2199      2182 2      SEARCH_FLAGS = QUI$M SEARCH WILDCARD;
: 2200      2183 2      IF CLISP$PRESENT($DESCRIPTOR('ALL'))
: 2201      2184 2      THEN SEARCH_FLAGS[QUI$V_SEARCH_ALL_JOBS] = TRUE;
: 2202      2185 2      JOB_PRINTED = FALSE;
: 2203      2186
: 2204      2187
: 2205      2188 2      ! Initialize the item list.
: 2206      2189 2      !
: 2207      2190 2      ITMLST[0,0,16,0] = %ALLOCATION(SEARCH_FLAGS);
: 2208      2191 2      ITMLST[2,0,16,0] = QUI$ SEARCH_FLAGS;
: 2209      2192 2      ITMLST[4,0,32,0] = SEARCH_FLAGS;
: 2210      2193 2      ITMLST[8,0,32,0] = 0;
: 2211      2194
: 2212      2195 2      ITMLST[12,0,16,0] = %ALLOCATION(AFTER_TIME);
: 2213      2196 2      ITMLST[14,0,16,0] = QUI$ AFTER_TIME;
: 2214      2197 2      ITMLST[16,0,32,0] = AFTER_TIME;
: 2215      2198 2      ITMLST[20,0,32,0] = 0;
: 2216      2199
: 2217      2200 2      ITMLST[24,0,16,0] = %ALLOCATION(COMPLETED_BLOCKS);
: 2218      2201 2      ITMLST[26,0,16,0] = QUI$ COMPLETED_BLOCKS;
: 2219      2202 2      ITMLST[28,0,32,0] = COMPLETED_BLOCKS;
```



```
: 2220      2203 2 ITMLST[32,0,32,0] = 0;
: 2221      2204 2
: 2222      2205 2 ITMLST[36,0,16,0] = %ALLOCATION(CONDITION_VECTOR);
: 2223      2206 2 ITMLST[38,0,16,0] = QUI$ CONDITION_VECTOR;
: 2224      2207 2 ITMLST[40,0,32,0] = CONDITION_VECTOR;
: 2225      2208 2 ITMLST[44,0,32,0] = 0;
: 2226      2209 2
: 2227      2210 2 ITMLST[48,0,16,0] = %ALLOCATION(ENTRY_NUMBER);
: 2228      2211 2 ITMLST[50,0,16,0] = QUI$ ENTRY_NUMBER;
: 2229      2212 2 ITMLST[52,0,32,0] = ENTRY_NUMBER;
: 2230      2213 2 ITMLST[56,0,32,0] = 0;
: 2231      2214 2
: 2232      2215 2 ITMLST[60,0,16,0] = %ALLOCATION(INTERVENING_BLOCKS);
: 2233      2216 2 ITMLST[62,0,16,0] = QUI$ INTERVENING_BLOCKS;
: 2234      2217 2 ITMLST[64,0,32,0] = INTERVENING_BLOCKS;
: 2235      2218 2 ITMLST[68,0,32,0] = 0;
: 2236      2219 2
: 2237      2220 2 ITMLST[72,0,16,0] = %ALLOCATION(INTERVENING_JOBS);
: 2238      2221 2 ITMLST[74,0,16,0] = QUI$ INTERVENING_JOBS;
: 2239      2222 2 ITMLST[76,0,32,0] = INTERVENING_JOBS;
: 2240      2223 2 ITMLST[80,0,32,0] = 0;
: 2241      2224 2
: 2242      2225 2 ITMLST[84,0,16,0] = %ALLOCATION(JOB_COPIES);
: 2243      2226 2 ITMLST[86,0,16,0] = QUI$ JOB_COPIES;
: 2244      2227 2 ITMLST[88,0,32,0] = JOB_COPIES;
: 2245      2228 2 ITMLST[92,0,32,0] = 0;
: 2246      2229 2
: 2247      2230 2 ITMLST[96,0,16,0] = %ALLOCATION(JOB_COPIES_DONE);
: 2248      2231 2 ITMLST[98,0,16,0] = QUI$ JOB_COPIES_DONE;
: 2249      2232 2 ITMLST[100,0,32,0] = JOB_COPIES_DONE;
: 2250      2233 2 ITMLST[104,0,32,0] = 0;
: 2251      2234 2
: 2252      2235 2 ITMLST[108,0,16,0] = %ALLOCATION(JOB_FLAGS);
: 2253      2236 2 ITMLST[110,0,16,0] = QUI$ JOB_FLAGS;
: 2254      2237 2 ITMLST[112,0,32,0] = JOB_FLAGS;
: 2255      2238 2 ITMLST[116,0,32,0] = 0;
: 2256      2239 2
: 2257      2240 2 ITMLST[120,0,16,0] = %ALLOCATION(JOB_NAME);
: 2258      2241 2 ITMLST[122,0,16,0] = QUI$ JOB_NAME;
: 2259      2242 2 ITMLST[124,0,32,0] = JOB_NAME;
: 2260      2243 2 ITMLST[128,0,32,0] = JOB_NAME_LENGTH;
: 2261      2244 2
: 2262      2245 2 ITMLST[132,0,16,0] = %ALLOCATION(JOB_SIZE);
: 2263      2246 2 ITMLST[134,0,16,0] = QUI$ JOB_SIZE;
: 2264      2247 2 ITMLST[136,0,32,0] = JOB_SIZE;
: 2265      2248 2 ITMLST[140,0,32,0] = 0;
: 2266      2249 2
: 2267      2250 2 ITMLST[144,0,16,0] = %ALLOCATION(JOB_STATUS);
: 2268      2251 2 ITMLST[146,0,16,0] = QUI$ JOB_STATUS;
: 2269      2252 2 ITMLST[148,0,32,0] = JOB_STATUS;
: 2270      2253 2 ITMLST[152,0,32,0] = 0;
: 2271      2254 2
: 2272      2255 2 ITMLST[156,0,16,0] = %ALLOCATION(REFUSAL_REASON);
: 2273      2256 2 ITMLST[158,0,16,0] = QUI$ REFUSAL_REASON;
: 2274      2257 2 ITMLST[160,0,32,0] = REFUSAL_REASON;
: 2275      2258 2 ITMLST[164,0,32,0] = REFUSAL_REASON_LENGTH;
: 2276      2259 2
```

```
2277 2260 2 ITMLST[168.0,16.0] = %ALLOCATION(USERNAME);
2278 2261 2 ITMLST[170.0,16.0] = QUI$ USERNAME;
2279 2262 2 ITMLST[172.0,32.0] = USERNAME;
2280 2263 2 ITMLST[176.0,32.0] = USERNAME_LENGTH;
2281 2264 2
2282 2265 2 ITMLST[180.0,32.0] = 0;
2283 2266 2
2284 2267 2
2285 2268 2 IF .FLAGS[V_FULL]
2286 2269 2 THEN
2287 2270 2 BEGIN
2288 2271 2 ITMLST[180.0,16.0] = %ALLOCATION(CHARACTERISTICS);
2289 2272 2 ITMLST[182.0,16.0] = QUI$ CHARACTERISTICS;
2290 2273 2 ITMLST[184.0,32.0] = CHARACTERISTICS;
2291 2274 2 ITMLST[188.0,32.0] = 0;
2292 2275 2
2293 2276 2 ITMLST[192.0,16.0] = %ALLOCATION(CHECKPOINT_DATA);
2294 2277 2 ITMLST[194.0,16.0] = QUI$ CHECKPOINT_DATA;
2295 2278 2 ITMLST[196.0,32.0] = CHECKPOINT_DATA;
2296 2279 2 ITMLST[200.0,32.0] = CHECKPOINT_DATA_LENGTH;
2297 2280 2
2298 2281 2 ITMLST[204.0,16.0] = %ALLOCATION(CLI);
2299 2282 2 ITMLST[206.0,16.0] = QUI$ CLI;
2300 2283 2 ITMLST[208.0,32.0] = CLI;
2301 2284 2 ITMLST[212.0,32.0] = CLI_LENGTH;
2302 2285 2
2303 2286 2 ITMLST[216.0,16.0] = %ALLOCATION(CPU_LIMIT);
2304 2287 2 ITMLST[218.0,16.0] = QUI$ CPU_LIMIT;
2305 2288 2 ITMLST[220.0,32.0] = CPU_LIMIT;
2306 2289 2 ITMLST[224.0,32.0] = 0;
2307 2290 2
2308 2291 2 ITMLST[228.0,16.0] = %ALLOCATION(FORM_NAME);
2309 2292 2 ITMLST[230.0,16.0] = QUI$ FORM_NAME;
2310 2293 2 ITMLST[232.0,32.0] = FORM_NAME;
2311 2294 2 ITMLST[236.0,32.0] = FORM_NAME_LENGTH;
2312 2295 2
2313 2296 2 ITMLST[240.0,16.0] = %ALLOCATION(LOG_QUEUE);
2314 2297 2 ITMLST[242.0,16.0] = QUI$ LOG_QUEUE;
2315 2298 2 ITMLST[244.0,32.0] = LOG_QUEUE;
2316 2299 2 ITMLST[248.0,32.0] = LOG_QUEUE_LENGTH;
2317 2300 2
2318 2301 2 ITMLST[252.0,16.0] = %ALLOCATION(LOG_SPECIFICATION);
2319 2302 2 ITMLST[254.0,16.0] = QUI$ LOG_SPECIFICATION;
2320 2303 2 ITMLST[256.0,32.0] = LOG_SPECIFICATION;
2321 2304 2 ITMLST[260.0,32.0] = LOG_SPECIFICATION_LENGTH;
2322 2305 2
2323 2306 2 ITMLST[264.0,16.0] = %ALLOCATION(NOTE);
2324 2307 2 ITMLST[266.0,16.0] = QUI$ NOTE;
2325 2308 2 ITMLST[268.0,32.0] = NOTE;
2326 2309 2 ITMLST[272.0,32.0] = NOTE_LENGTH;
2327 2310 2
2328 2311 2 ITMLST[276.0,16.0] = %ALLOCATION(OPERATOR_REQUEST);
2329 2312 2 ITMLST[278.0,16.0] = QUI$ OPERATOR_REQUEST;
2330 2313 2 ITMLST[280.0,32.0] = OPERATOR_REQUEST;
2331 2314 2 ITMLST[284.0,32.0] = OPERATOR_REQUEST_LENGTH;
2332 2315 2
2333 2316 2 ITMLST[288.0,16.0] = %ALLOCATION(PARAMETER_1);
```



```
2334 2317 ITMLST[290.0,16.0] = QUIS_PARAMETER_1;
2335 2318 ITMLST[292.0,32.0] = PARAMETER_1;
2336 2319 ITMLST[296.0,32.0] = PARAMETER_1_LENGTH;
2337 2320
2338 2321 ITMLST[300.0,16.0] = %ALLOCATION(PARAMETER_2);
2339 2322 ITMLST[302.0,16.0] = QUIS_PARAMETER_2;
2340 2323 ITMLST[304.0,32.0] = PARAMETER_2;
2341 2324 ITMLST[308.0,32.0] = PARAMETER_2_LENGTH;
2342 2325
2343 2326 ITMLST[312.0,16.0] = %ALLOCATION(PARAMETER_3);
2344 2327 ITMLST[314.0,16.0] = QUIS_PARAMETER_3;
2345 2328 ITMLST[316.0,32.0] = PARAMETER_3;
2346 2329 ITMLST[320.0,32.0] = PARAMETER_3_LENGTH;
2347 2330
2348 2331 ITMLST[324.0,16.0] = %ALLOCATION(PARAMETER_4);
2349 2332 ITMLST[326.0,16.0] = QUIS_PARAMETER_4;
2350 2333 ITMLST[328.0,32.0] = PARAMETER_4;
2351 2334 ITMLST[332.0,32.0] = PARAMETER_4_LENGTH;
2352 2335
2353 2336 ITMLST[336.0,16.0] = %ALLOCATION(PARAMETER_5);
2354 2337 ITMLST[338.0,16.0] = QUIS_PARAMETER_5;
2355 2338 ITMLST[340.0,32.0] = PARAMETER_5;
2356 2339 ITMLST[344.0,32.0] = PARAMETER_5_LENGTH;
2357 2340
2358 2341 ITMLST[348.0,16.0] = %ALLOCATION(PARAMETER_6);
2359 2342 ITMLST[350.0,16.0] = QUIS_PARAMETER_6;
2360 2343 ITMLST[352.0,32.0] = PARAMETER_6;
2361 2344 ITMLST[356.0,32.0] = PARAMETER_6_LENGTH;
2362 2345
2363 2346 ITMLST[360.0,16.0] = %ALLOCATION(PARAMETER_7);
2364 2347 ITMLST[362.0,16.0] = QUIS_PARAMETER_7;
2365 2348 ITMLST[364.0,32.0] = PARAMETER_7;
2366 2349 ITMLST[368.0,32.0] = PARAMETER_7_LENGTH;
2367 2350
2368 2351 ITMLST[372.0,16.0] = %ALLOCATION(PARAMETER_8);
2369 2352 ITMLST[374.0,16.0] = QUIS_PARAMETER_8;
2370 2353 ITMLST[376.0,32.0] = PARAMETER_8;
2371 2354 ITMLST[380.0,32.0] = PARAMETER_8_LENGTH;
2372 2355
2373 2356 ITMLST[384.0,16.0] = %ALLOCATION(PRIORITY);
2374 2357 ITMLST[386.0,16.0] = QUIS_PRIORITY;
2375 2358 ITMLST[388.0,32.0] = PRIORITY;
2376 2359 ITMLST[392.0,32.0] = 0;
2377 2360
2378 2361 ITMLST[396.0,16.0] = %ALLOCATION(SUBMISSION_TIME);
2379 2362 ITMLST[398.0,16.0] = QUIS_SUBMISSION_TIME;
2380 2363 ITMLST[400.0,32.0] = SUBMISSION_TIME;
2381 2364 ITMLST[404.0,32.0] = 0;
2382 2365
2383 2366 ITMLST[408.0,16.0] = %ALLOCATION(WDEFAULT);
2384 2367 ITMLST[410.0,16.0] = QUIS_WDEFAULT;
2385 2368 ITMLST[412.0,32.0] = WDEFAULT;
2386 2369 ITMLST[416.0,32.0] = 0;
2387 2370
2388 2371 ITMLST[420.0,16.0] = %ALLOCATION(WEXTENT);
2389 2372 ITMLST[422.0,16.0] = QUIS_WEXTENT;
2390 2373 ITMLST[424.0,32.0] = WEXTENT;
```

```
2391 2374 3 ITMLST[428,0,32,0] = 0;
2392 2375 3
2393 2376 3 ITMLST[432,0,16,0] = %ALLOCATION(WSQUOTA);
2394 2377 3 ITMLST[434,0,16,0] = QUIS WSQUOTA;
2395 2378 3 ITMLST[436,0,32,0] = WSQUOTA;
2396 2379 3 ITMLST[440,0,32,0] = 0;
2397 2380 3
2398 2381 3 ITMLST[444,0,32,0] = 0;
2399 2382 3 END;
2400 2383 3
2401 2384 3
2402 2385 3 ! Loop to get all jobs.
2403 2386 3
2404 2387 3 WHILE TRUE DO
2405 2388 3 BEGIN
2406 2389 3 STATUS = $GETQUIW(
2407 2390 3 FUNC=QUIS DISPLAY_JOB,
2408 2391 3 ITMLST=ITMLST,
2409 2392 3 IOSB=IOSB);
2410 2393 3 IF .STATUS THEN STATUS = .IOSB;
2411 2394 3 IF NOT .STATUS THEN EXITLOOP;
2412 2395 3
2413 2396 3
2414 2397 3 FLAGS[V TRUNCATE] = TRUE;
2415 2398 3 IF TESTBITCS(JOB_PRINTED)
2416 2399 3 THEN
2417 2400 3 BEGIN
2418 2401 3 FAO (' Jobname Username Entry');
2419 2402 3 IF NOT .QUEUE_FLAGS[QUISV_QUEUE_BATCH] THEN FAO (' Blocks');
2420 2403 3 FAOL (' Status');
2421 2404 3
2422 2405 3 FAO (' -----');
2423 2406 3 IF NOT .QUEUE_FLAGS[QUISV_QUEUE_BATCH] THEN FAO (' -----');
2424 2407 3 FAOL (' -----');
2425 2408 3 END;
2426 2409 3
2427 2410 3
2428 2411 3 IF .INTERVENING_JOBS NEQ 0
2429 2412 3 THEN
2430 2413 3 BEGIN
2431 2414 3 FAO (' (!UL intervening job!%S', .INTERVENING_JOBS);
2432 2415 3 IF .INTERVENING_BLOCKS NEQ 0
2433 2416 3 THEN FAO ('-containing !UL block!%S', .INTERVENING_BLOCKS);
2434 2417 3 FAOL (' ');
2435 2418 3 END;
2436 2419 3
2437 2420 3
2438 2421 3 IF NOT .JOB_STATUS[QUISV_JOB_INACCESSIBLE]
2439 2422 3 THEN
2440 2423 3 BEGIN
2441 2424 3 IF .JOB_NAME_LENGTH GTRU 15
2442 2425 3 THEN
2443 2426 3 BEGIN
2444 2427 3 FAOL (' !AF', .JOB_NAME_LENGTH, JOB_NAME);
2445 2428 3 FAO (' !17* ');
2446 2429 3 END
2447 2430 3 ELSE
```



```
2448      2431 5      BEGIN
2449      2432 5      FAO (' !15AF', .JOB_NAME_LENGTH, JOB_NAME);
2450      2433 4      END;
2451      2434 4
2452      2435 4      FAO (' !12AF', .USERNAME_LENGTH, USERNAME);
2453      2436 4      END
2454      2437 3      ELSE
2455      2438 4      BEGIN
2456      2439 4      FAO (' no privilege!16* ')
2457      2440 3      END;
2458      2441 3      FAO (' !5UL', .ENTRY_NUMBER);
2459      2442 3      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
2460      2443 3      THEN FAO_T ' !6UL', .JOB_SIZE);
2461      2444 3
2462      2445 3
2463      2446 3      IF .JOB_STATUS[QUI$V_JOB_ABORTING]
2464      2447 3      THEN
2465      2448 3      FAO (' Aborting')
2466      2449 3
2467      2450 3
2468      2451 3      ELSE IF .JOB_STATUS[QUI$V_JOB_EXECUTING]
2469      2452 3      THEN
2470      2453 4      BEGIN
2471      2454 4      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
2472      2455 4      THEN
2473      2456 4      FAO (' Executing')
2474      2457 4      ELSE IF .QUEUE_STATUS[QUI$V_QUEUE_SERVER]
2475      2458 4      THEN
2476      2459 4      FAO (' Processing')
2477      2460 4      ELSE
2478      2461 5      BEGIN
2479      2462 5      FAO (' Printing');
2480      2463 5      IF .JOB_COPIES_GTRU 1
2481      2464 5      THEN FAO (' job copy !UL', .JOB_COPIES_DONE + 1);
2482      2465 4      END;
2483      2466 4      END
2484      2467 4
2485      2468 4
2486      2469 3      ELSE IF .JOB_STATUS[QUI$V_JOB_HOLDING]
2487      2470 3      THEN
2488      2471 3      FAO (' Holding')
2489      2472 3
2490      2473 3
2491      2474 3      ELSE IF .JOB_STATUS[QUI$V_JOB_REFUSED]
2492      2475 3      THEN
2493      2476 4      BEGIN
2494      2477 4      FAO (' Waiting');
2495      2478 4      IF .REFUSAL_REASON_LENGTH NEQ 0
2496      2479 4      THEN
2497      2480 5      BEGIN
2498      2481 5      IF .COMPLETED_BLOCKS NEQ 0
2499      2482 5      THEN
2500      2483 6      BEGIN
2501      2484 6      FAO (' at block !UL', .COMPLETED_BLOCKS);
2502      2485 6      COMPLETED_BLOCKS = 0;
2503      2486 5      END;
2504      2487 5      EOL();
```

```
2505      2488 5      FAO(' !AF', .REFUSAL_REASON_LENGTH, REFUSAL_REASON);
2506      2489 4      END;
2507      2490 4      END
2508      2491 4
2509      2492 4
2510      2493 3      ELSE IF .JOB_STATUS[QUI$V_JOB_RETAINED]
2511      2494 3      THEN
2512      2495 4      BEGIN
2513      2496 4      FAO(' Completed');
2514      2497 4      IF .CONDITION_VECTOR[0] NEQ 0
2515      2498 4      AND NOT .CONDITION_VECTOR[0]
2516      2499 4      THEN
2517      2500 5      BEGIN
2518      2501 5      INCR I FROM 0 TO 2 DO
2519      2502 6      BEGIN
2520      2503 6      IF .CONDITION_VECTOR[I] NEQ 0
2521      2504 6      THEN
2522      2505 7      BEGIN
2523      2506 7      LOCAL
2524      2507 7      MESSAGE_DESC: VECTOR[2],
2525      2508 7      MESSAGE: VECTOR[132,BYTE];
2526      2509 7
2527      2510 7      MESSAGE_DESC[0] = %ALLOCATION(MESSAGE);
2528      2511 7      MESSAGE_DESC[1] = MESSAGE;
2529      2512 7      $GETMSGT
2530      2513 7      MSGID=.CONDITION_VECTOR[I],
2531      2514 7      MSGLEN=MESSAGE_DESC,
2532      2515 7      BUFADR=MESSAGE_DESC,
2533      2516 7      FLAGS=%B'1111';
2534      2517 7      IF .I NEQ 0 THEN MESSAGE[0] = %C'-' ;
2535      2518 7      IF .COMPLETED_BLOCKS NEQ 0
2536      2519 7      THEN
2537      2520 8      BEGIN
2538      2521 8      FA (' at block !UL', .COMPLETED_BLOCKS);
2539      2522 8      COMPLETED_BLOCKS = 0;
2540      2523 7      END;
2541      2524 7      EOL();
2542      2525 7      FAO(' !AF',
2543      2526 7      .MESSAGE_DESC[0], .MESSAGE_DESC[1]);
2544      2527 6      END;
2545      2528 5      END;
2546      2529 4      END;
2547      2530 4      END
2548      2531 4
2549      2532 4
2550      2533 3      ELSE IF .JOB_STATUS[QUI$V_JOB_STARTING]
2551      2534 3      THEN
2552      2535 3      FAO(' Starting')
2553      2536 3
2554      2537 3
2555      2538 3      ELSE IF .JOB_STATUS[QUI$V_JOB_TIMED]
2556      2539 3      THEN
2557      2540 3      FAO(' Holding until !17%D', AFTER_TIME)
2558      2541 3
2559      2542 3
2560      2543 3      ELSE
2561      2544 3      FAO(' Pending');
```



```
2562 2545 3
2563 2546 3
2564 2547 3
2565 2548 3
2566 2549 3
2567 2550 3
2568 2551 3
2569 2552 3
2570 2553 3
2571 2554 3
2572 2555 3
2573 2556 3
2574 2557 3
2575 2558 4
2576 2559 4
2577 2560 4
2578 2561 5
2579 2562 5
2580 2563 5
2581 2564 5
2582 2565 5
2583 2566 5
2584 2567 5
2585 2568 5
2586 2569 5
2587 2570 5
2588 2571 5
2589 2572 6
2590 2573 6
2591 2574 6
2592 2575 7
2593 2576 7
2594 2577 7
2595 2578 7
2596 2579 7
2597 2580 7
2598 2581 7
2599 2582 6
2600 2583 5
2601 2584 5
2602 2585 5
2603 2586 5
2604 2587 5
2605 2588 5
2606 2589 5
2607 2590 5
2608 2591 6
2609 2592 6
2610 2593 6
2611 2594 7
2612 2595 7
2613 2596 7
2614 2597 6
2615 2598 6
2616 2599 6
2617 2600 6
2618 2601 6

IF .COMPLETED_BLOCKS NEQ 0
THEN
    FAO (' at block !UL', .COMPLETED_BLOCKS);

EOL ();
FLAGS[V_TRUNCATE] = FALSE;

IF NOT .JOB_STATUS[QUI$V_JOB_INACCESSIBLE]
THEN
    BEGIN
        IF .FLAGS[V_FULL]
        THEN
            BEGIN
                FLAGS[B_LEFT_MARGIN] = 4;
                FLAGS[B_PENDING_SPACES] = 4;

                FAO ('Submitted !17%D', SUBMISSION_TIME);
                FLAGS[B_PENDING_SPACES] = 1;

                IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
                THEN
                    BEGIN
                        IF .JOB_FLAGS[QUI$V_JOB_FILE_BURST_EXP]
                        THEN
                            BEGIN
                                IF .JOB_FLAGS[QUI$V_JOB_FILE_BURST]
                                THEN FAO ('/BURST')
                                ELSE IF .JOB_FLAGS[QUI$V_JOB_FILE_BURST_ONE]
                                THEN FAO ('/BURST=ONE')
                                ELSE FAO ('/NOBURST');
                                FLAGS[B_PENDING_SPACES] = 1;
                            END;
                        END;
                    END;

                FORMAT_CHARACTERISTICS(CCHARACTERISTICS);

                IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
                THEN
                    BEGIN
                        IF .CLI_LENGTH NEQ 0
                        THEN
                            BEGIN
                                FAO ('/CLI=!AF', .CLI_LENGTH, CLI);
                                FLAGS[B_PENDING_SPACES] = 1;
                            END;

                        IF .JOB_FLAGS[QUI$V_JOB_CPU_LIMIT]
                        THEN
```

```
: 2619      2602  6      FORMAT CPU TIME(
: 2620      2603  6          UP[IT BYTE (%ASCIC 'CPU'),
: 2621      2604  6          .CPU_LIMIT);
: 2622      2605  6
: 2623      2606  6
: 2624      2607  6      IF NOT .JOB_FLAGS[QUI$V_JOB_LOG_DELETE]
: 2625      2608  6      THEN
: 2626      2609  7          BEGIN
: 2627      2610  7              FAO ('/KEEP');
: 2628      2611  7              FLAGS[B_PENDING_SPACES] = 1;
: 2629      2612  6              END;
: 2630      2613  6
: 2631      2614  6
: 2632      2615  6      IF .JOB_FLAGS[QUI$V_JOB_LOG_NULL]
: 2633      2616  6      THEN
: 2634      2617  7          BEGIN
: 2635      2618  7              FAO ('/NOLOG');
: 2636      2619  7              FLAGS[B_PENDING_SPACES] = 1;
: 2637      2620  6              END;
: 2638      2621  6
: 2639      2622  6
: 2640      2623  6      IF .LOG_SPECIFICATION_LENGTH NEQ 0
: 2641      2624  6      THEN
: 2642      2625  7          BEGIN
: 2643      2626  7              FAO ('/LOG=!AF',
: 2644      2627  7                  .LOG_SPECIFICATION_LENGTH, LOG_SPECIFICATION);
: 2645      2628  7              FLAGS[B_PENDING_SPACES] = 1;
: 2646      2629  6              END;
: 2647      2630  6      END
: 2648      2631  5      ELSE
: 2649      2632  6          BEGIN
: 2650      2633  6              IF .JOB_FLAGS[QUI$V_JOB_FILE_FLAG_EXP]
: 2651      2634  6              THEN
: 2652      2635  7                  BEGIN
: 2653      2636  7                      IF .JOB_FLAGS[QUI$V_JOB_FILE_FLAG]
: 2654      2637  7                      THEN FAO ('/FLAG');
: 2655      2638  7                      ELSE IF .JOB_FLAGS[QUI$V_JOB_FILE_FLAG_ONE]
: 2656      2639  7                      THEN FAO ('/FLAG=ONE');
: 2657      2640  7                      ELSE FAO ('/NOFLAG');
: 2658      2641  7                      FLAGS[B_PENDING_SPACES] = 1;
: 2659      2642  6                      END;
: 2660      2643  6
: 2661      2644  6
: 2662      2645  6      IF .JOB_FLAGS[QUI$V_JOB_FILE_PAGINATE_EXP]
: 2663      2646  6      THEN
: 2664      2647  7          BEGIN
: 2665      2648  7              IF .JOB_FLAGS[QUI$V_JOB_FILE_PAGINATE]
: 2666      2649  7              THEN FAO ('/FEED');
: 2667      2650  7              ELSE FAO ('/NOFEED');
: 2668      2651  7              FLAGS[B_PENDING_SPACES] = 1;
: 2669      2652  6              END;
: 2670      2653  6
: 2671      2654  6
: 2672      2655  6      FAO ('/FORM=!AF', .FORM_NAME_LENGTH, FORM_NAME);
: 2673      2656  6      FLAGS[B_PENDING_SPACES] = 1;
: 2674      2657  6
: 2675      2658  6
```



```
: 2676      2659  6      IF .JOB_COPIES GTRU 1
: 2677      2660  6      THEN
: 2678      2661  7          BEGIN
: 2679      2662  7              FAO ('/JOB COUNT='!UL', .JOB_COPIES);
: 2680      2663  7              FLAGS[B_PENDING_SPACES] = 1;
: 2681      2664  6              END;
: 2682      2665  6
: 2683      2666  6
: 2684      2667  6      IF .JOB_FLAGS[QUI$V_JOB_LOWERCASE]
: 2685      2668  6      THEN
: 2686      2669  7          BEGIN
: 2687      2670  7              FAO ('/LOWER');
: 2688      2671  7              FLAGS[B_PENDING_SPACES] = 1;
: 2689      2672  6              END;
: 2690      2673  6
: 2691      2674  6
: 2692      2675  6      IF .NOTE_LENGTH NEQ 0
: 2693      2676  6      THEN
: 2694      2677  7          BEGIN
: 2695      2678  7              FAO ('/NOTE='!AF'', .NOTE_LENGTH, NOTE);
: 2696      2679  7              FLAGS[B_PENDING_SPACES] = 1;
: 2697      2680  6              END;
: 2698      2681  5      END;
: 2699      2682  5
: 2700      2683  5
: 2701      2684  5      IF .JOB_FLAGS[QUI$V_JOB_NOTIFY]
: 2702      2685  5      THEN
: 2703      2686  6          BEGIN
: 2704      2687  6              FAO ('/NOTIFY');
: 2705      2688  6              FLAGS[B_PENDING_SPACES] = 1;
: 2706      2689  5              END;
: 2707      2690  5
: 2708      2691  5
: 2709      2692  5      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 2710      2693  5      THEN
: 2711      2694  6          BEGIN
: 2712      2695  6              IF .OPERATOR_REQUEST_LENGTH NEQ 0
: 2713      2696  6              THEN
: 2714      2697  7                  BEGIN
: 2715      2698  7                      FAO ('/OPERATOR='!AF'',
: 2716      2699  7                          .OPERATOR_REQUEST_LENGTH, OPERATOR_REQUEST);
: 2717      2700  7                      FLAGS[B_PENDING_SPACES] = 1;
: 2718      2701  6                      END;
: 2719      2702  5                  END;
: 2720      2703  5
: 2721      2704  5
: 2722      2705  5      FORMAT PARAMETERS(
: 2723      2706  5          .PARAMETER_1_LENGTH, PARAMETER_1,
: 2724      2707  5          .PARAMETER_2_LENGTH, PARAMETER_2,
: 2725      2708  5          .PARAMETER_3_LENGTH, PARAMETER_3,
: 2726      2709  5          .PARAMETER_4_LENGTH, PARAMETER_4,
: 2727      2710  5          .PARAMETER_5_LENGTH, PARAMETER_5,
: 2728      2711  5          .PARAMETER_6_LENGTH, PARAMETER_6,
: 2729      2712  5          .PARAMETER_7_LENGTH, PARAMETER_7,
: 2730      2713  5          .PARAMETER_8_LENGTH, PARAMETER_8);
: 2731      2714  5
: 2732      2715  5
```

```
: 2733      2716 5      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 2734      2717 5      THEN
: 2735      2718 6      BEGIN
: 2736      2719 6      IF NOT .JOB_FLAGS[QUI$V_JOB_LOG_SPOOL]
: 2737      2720 6      THEN
: 2738      2721 7      BEGIN
: 2739      2722 7      FAO ('/NOPRINT');
: 2740      2723 7      FLAGS[B_PENDING_SPACES] = 1;
: 2741      2724 6      END;
: 2742      2725 6
: 2743      2726 6
: 2744      2727 6      IF .LOG_QUEUE_LENGTH NEQ 0
: 2745      2728 6      THEN
: 2746      2729 7      BEGIN
: 2747      2730 7      FAO ('/PRINT=!AF', .LOG_QUEUE_LENGTH, LOG_QUEUE);
: 2748      2731 7      FLAGS[B_PENDING_SPACES] = 1;
: 2749      2732 6      END;
: 2750      2733 5      END;
: 2751      2734 5
: 2752      2735 5
: 2753      2736 5      FAO ('/PRIO=!UL', .PRIORITY);
: 2754      2737 5      FLAGS[B_PENDING_SPACES] = 1;
: 2755      2738 5
: 2756      2739 5
: 2757      2740 5
: 2758      2741 5      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
: 2759      2742 6      THEN
: 2760      2743 6      BEGIN
: 2761      2744 6      IF .JOB_FLAGS[QUI$V_JOB_RESTART]
: 2762      2745 7      THEN
: 2763      2746 7      BEGIN
: 2764      2747 7      FAO ('/RESTART');
: 2765      2748 6      FLAGS[B_PENDING_SPACES] = 1;
: 2766      2749 6      END;
: 2767      2750 6
: 2768      2751 6
: 2769      2752 6      IF .CHECKPOINT_DATA_LENGTH NEQ 0
: 2770      2753 7      THEN
: 2771      2754 7      BEGIN
: 2772      2755 7      FAO ('RESTART VALUE="!AF"',
: 2773      2756 7      .CHECKPOINT_DATA_LENGTH, CHECKPOINT_DATA);
: 2774      2757 6      FLAGS[B_PENDING_SPACES] = 1;
: 2775      2758 6      END;
: 2776      2759 6
: 2777      2760 6
: 2778      2761 6      IF .JOB_FLAGS[QUI$V_JOB_WSDEFAULT]
: 2779      2762 6      THEN
: 2780      2763 6      FORMAT WORKING_SET(
: 2781      2764 6      UP[IT BYTE-(%ASCIC 'WSDEFAULT'),
: 2782      2765 6      .WSDEFAULT);
: 2783      2766 6
: 2784      2767 6
: 2785      2768 6      IF .JOB_FLAGS[QUI$V_JOB_WSEXTENT]
: 2786      2769 6      THEN
: 2787      2770 6      FORMAT WORKING_SET(
: 2788      2771 6      UP[IT BYTE-(%ASCIC 'WSEXTENT'),
: 2789      2772 6      .WSEXTENT);
```


												4C	4C	41	01669	P.AEX:	.ASCII	\ALL\							
												00000003	0166C	P.AEW:	.LONG	3									
												00000000	01670		.ADDRESS	P.AEX									
20	20	20	20	20	65	6D	61	6E	62	6F	4A	20	20	24	01674	P.AEY:	.ASCII	\\$	Jobname	Username	Entry\				
20	20	20	65	6D	61	6E	72	65	73	55	20	20	20	20	01683										
												79	72	74	6E	45	20	20	01692						
												73	6B	63	6F	6C	42	20	20	08	01699	P.AEZ:	.ASCII	<8>\	Blocks\
												73	75	74	61	74	53	20	20	08	016A2	P.AFA:	.ASCII	<8>\	Status\
20	20	20	20	20	2D	2D	2D	2D	2D	2D	2D	20	20	24	016AB	P.AFB:	.ASCII	\\$	-----	-----	-----\				
20	20	20	2D	2D	2D	2D	2D	2D	2D	2D	20	20	20	20	016BA										

[illegible]


```

.EXTRN  SYS$GETMSG

```

Address	Hex	ASCII	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	
---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

03	F9	A9	01	C6	31	00173	2\$:	MOVAB	#2883623, ITMLST+120	2240
								MOVAB	JOB_NAME, ITMLST+124	2242
								MOVAB	JOB_NAME_LENGTH, ITMLST+128	2243
								MOVAB	JOB_SIZE, ITMLST+136	2247
								CLRL	ITMLST+140	2248
								MOVAB	JOB_STATUS, ITMLST+148	2252
								CLRL	ITMLST+152	2253
								MOVAB	REFUSAL_REASON, ITMLST+160	2257
								MOVAB	REFUSAL_REASON_LENGTH, ITMLST+164	2258
								MOVAB	USERNAME, ITMLST+172	2262
								MOVAB	USERNAME_LENGTH, ITMLST+176	2263
								CLRL	ITMLST+180	2265
								BBS	#1, FLAGS, 2\$	2268
								BRW	3\$	2271
								MOVAB	CHARACTERISTICS, ITMLST+184	2273
								CLRL	ITMLST+188	2274
								MOVAB	CHECKPOINT_DATA, ITMLST+196	2278
								MOVAB	CHECKPOINT_DATA_LENGTH, ITMLST+200	2279
								MOVAB	CLI, ITMLST+208	2283
								MOVAB	CLI_LENGTH, ITMLST+212	2284
								MOVAB	CPU_LIMIT, ITMLST+220	2288
								CLRL	ITMLST+224	2289
								MOVAB	FORM_NAME, ITMLST+232	2293
								MOVAB	FORM_NAME_LENGTH, ITMLST+236	2294
								MOVAB	LOG_QUEUE, ITMLST+244	2298
								MOVAB	LOG_QUEUE_LENGTH, ITMLST+248	2299
								MOVAB	LOG_SPECIFICATION, ITMLST+256	2303
								MOVAB	LOG_SPECIFICATION_LENGTH, ITMLST+260	2304
								MOVAB	NOTE, ITMLST+268	2308
								MOVAB	NOTE_LENGTH, ITMLST+272	2309
								MOVAB	OPERATOR_REQUEST, ITMLST+280	2313
								MOVAB	OPERATOR_REQUEST_LENGTH, ITMLST+284	2314
								MOVAB	PARAMETER_1, ITMLST+292	2318
								MOVAB	PARAMETER_1_LENGTH, ITMLST+296	2319
								MOVAB	PARAMETER_2, ITMLST+304	2323
								MOVAB	PARAMETER_2_LENGTH, ITMLST+308	2324
								MOVAB	PARAMETER_3, ITMLST+316	2328
								MOVAB	PARAMETER_3_LENGTH, ITMLST+320	2329
								MOVAB	PARAMETER_4, ITMLST+328	2333
								MOVAB	PARAMETER_4_LENGTH, ITMLST+332	2334

90	AD	003E00FF	8F	DO	00297	MOVL	#4063487, ITMLST+336	2336
94	AD	04D4	CE	9E	0029F	MOVAB	PARAMETER_5, ITMLST+340	2338
98	AD	04D2	CE	9E	002A5	MOVAB	PARAMETER_5 LENGTH, ITMLST+344	2339
9C	AD	003F00FF	8F	DO	002AB	MOVL	#4129023, ITMLST+348	2341
A0	AD	03D0	CE	9E	002B3	MOVAB	PARAMETER_6, ITMLST+352	2343
A4	AD	03CE	CE	9E	002B9	MOVAB	PARAMETER_6 LENGTH, ITMLST+356	2344
A8	AD	004000FF	8F	DO	002BF	MOVL	#4194559, ITMLST+360	2346
AC	AD	02CC	CE	9E	002C7	MOVAB	PARAMETER_7, ITMLST+364	2348
B0	AD	02CA	CE	9E	002CD	MOVAB	PARAMETER_7 LENGTH, ITMLST+368	2349
B4	AD	004100FF	8F	DO	002D3	MOVL	#4260095, ITMLST+372	2351
B8	AD	01C8	CE	9E	002DB	MOVAB	PARAMETER_8, ITMLST+376	2353
BC	AD	01C6	CE	9E	002E1	MOVAB	PARAMETER_8 LENGTH, ITMLST+380	2354
CO	AD	00420004	8F	DO	002E7	MOVL	#4325380, ITMLST+384	2356
C4	AD	01C0	CE	9E	002EF	MOVAB	PRIORITY, ITMLST+388	2358
		C8	AD	D4	002F5	CLRL	ITMLST+392	2359
CC	AD	004F0008	8F	DO	002F8	MOVL	#5177352, ITMLST+396	2361
DO	AD	00B0	CE	9E	00300	MOVAB	SUBMISSION_TIME, ITMLST+400	2363
		D4	AD	D4	00306	CLRL	ITMLST+404	2364
D8	AD	00520004	8F	DO	00309	MOVL	#5373956, ITMLST+408	2366
DC	AD	009C	CE	9E	00311	MOVAB	WSDEFAULT, ITMLST+412	2368
		E0	AD	D4	00317	CLRL	ITMLST+416	2369
E4	AD	00530004	8F	DO	0031A	MOVL	#5439492, ITMLST+420	2371
E8	AD	0098	CE	9E	00322	MOVAB	WSEXTENT, ITMLST+424	2373
		EC	AD	D4	00328	CLRL	ITMLST+428	2374
F0	AD	00540004	8F	DO	0032B	MOVL	#5505028, ITMLST+432	2376
F4	AD	0094	CE	9E	00333	MOVAB	WSQUOTA, ITMLST+436	2378
		F8	AD	7C	00339	CLRL	ITMLST+440	2379
54	04	AC	01	00	EF 0033C	EXTZV	#0, #1, QUEUE_FLAGS, R4	2442
			55	54	D2 00342	MCOML	R4, R5	
				7E	7C 00345	CLRL	-(SP)	2392
		0094	CE	9F	00347	PUSHAB	IOSB	
		FE40	CD	9F	0034B	PUSHAB	ITMLST	
			7E	05	7D 0034F	MOVQ	#5, -(SP)	
				7E	D4 00352	CLRL	-(SP)	
00000000G			00	07	FB 00354	CALLS	#7, SYSSGETQUIW	
			53	50	DO 0035B	MOVL	R0, STATUS	
			05	53	E9 0035E	BLBC	STATUS, 5\$	2393
			53	CE	DO 00361	MOVL	IOSB, STATUS	
		008C	01	53	E8 00366	BLBS	STATUS, 6\$	2394
					04 00369	RET		
			F9	04	88 0036A	BISB2	#4, FLAGS	2397
38			56	00	E2 0036E	BBSS	#0, JOB_PRINTED, 9\$	2398
		F9AD		CF	9F 00372	PUSHAB	P.AEY	2401
			68	01	FB 00376	CALLS	#1, FAO	
		04	07	AC	E8 00379	BLBS	QUEUE_FLAGS, 7\$	2402
		F9C7		CF	9F 0037D	PUSHAB	P.AEZ	
			68	01	FB 00381	CALLS	#1, FAO	
		F9C9		CF	9F 00384	PUSHAB	P.AFA	2403
			68	01	FB 00388	CALLS	#1, FAO	
		6A		00	FB 0038B	CALLS	#0, EOL	
		F9C8		CF	9F 0038E	PUSHAB	P.AFB	2405
			68	01	FB 00392	CALLS	#1, FAO	
		04	07	AC	E8 00395	BLBS	QUEUE_FLAGS, 8\$	2406
		F9E2		CF	9F 00399	PUSHAB	P.AFC	
			68	01	FB 0039D	CALLS	#1, FAO	
		F9E4		CF	9F 003A0	PUSHAB	P.AFD	2407
			68	01	FB 003A4	CALLS	#1, FAO	

	6A		00	FB	003A7		CALLS	#0, EOL		
		FCB4	CD	D5	003AA	9\$:	TSTL	INTERVENING_JOBS	2411	
			26	13	003AE		BEQL	11\$		
		FCB4	CD	DD	003B0		PUSHL	INTERVENING_JOBS	2414	
		F9D9	CF	9F	003B4		PUSHAB	P.AFE		
	68		02	FB	003B8		CALLS	#2, FAO		
		FCB8	CD	D5	003BB		TSTL	INTERVENING_BLOCKS	2415	
			0B	13	003BF		BEQL	10\$		
		FCB8	CD	DD	003C1		PUSHL	INTERVENING_BLOCKS	2416	
		F9E4	CF	9F	003C5		PUSHAB	P.AFF		
	68		02	FB	003C9		CALLS	#2, FAO		
		F9F6	CF	9F	003CC	10\$:	PUSHAB	P.AFG	2417	
	68		01	FB	003D0		CALLS	#1, FAO		
	6A		00	FB	003D3		CALLS	#0, EOL		
45	FC74	CD	03	E0	003D6	11\$:	BBS	#3, JOB STATUS, 14\$	2421	
		OF		B1	003DC		CMPW	JOB_NAME_LENGTH, #15	2424	
			1C	1B	003E1		BLEQU	12\$		
		FC80	CD	9F	003E3		PUSHAB	JOB_NAME	2427	
	7E	FC7E	CD	3C	003E7		MOVZWL	JOB_NAME_LENGTH, -(SP)		
		F9D8	CF	9F	003EC		PUSHAB	P.AFH		
	68		03	FB	003F0		CALLS	#3, FAO		
	6A		00	FB	003F3		CALLS	#0, EOL		
		F9D4	CF	9F	003F6		PUSHAB	P.AFI	2428	
	68		01	FB	003FA		CALLS	#1, FAO		
			10	11	003FD		BRB	13\$	2424	
		FC80	CD	9F	003FF	12\$:	PUSHAB	JOB_NAME	2432	
	7E	FC7E	CD	3C	00403		MOVZWL	JOB_NAME_LENGTH, -(SP)		
		F9C8	CF	9F	00408		PUSHAB	P.AFJ		
	68		03	FB	0040C		CALLS	#3, FAO		
		00A4	CE	9F	0040F	13\$:	PUSHAB	USERNAME	2435	
	7E	00A6	CE	3C	00413		MOVZWL	USERNAME_LENGTH, -(SP)		
		F9C0	CF	9F	00418		PUSHAB	P.AFK		
	68		03	FB	0041C		CALLS	#3, FAO		
			07	11	0041F		BRB	15\$	2421	
		F9BE	CF	9F	00421	14\$:	PUSHAB	P.AFL	2439	
	68		01	FB	00425		CALLS	#1, FAO		
		FCE0	CD	DD	00428	15\$:	PUSHL	ENTRY_NUMBER	2441	
		F9C7	CF	9F	0042C		PUSHAB	P.AFM		
	68		02	FB	00430		CALLS	#2, FAO		
	0B		55	E9	00433		BLBC	R5, 16\$	2442	
		FC78	CD	DD	00436		PUSHL	JOB SIZE	2443	
		F9BF	CF	9F	0043A		PUSHAB	P.AFN		
	68		02	FB	0043E		CALLS	#2, FAO		
	06		CD	E9	00441	16\$:	BLBC	JOB STATUS, 17\$	2446	
		F9BA	CF	9F	00446		PUSHAB	P.AFO	2448	
			3F	11	0044A		BRB	21\$		
2F	FC74	CD	01	E1	0044C	17\$:	BBC	#1, JOB STATUS, 20\$	2451	
		06	54	E9	00452		BLBC	R4, 18\$	2454	
			CF	9F	00455		PUSHAB	P.AFP	2456	
		F9B6	30	11	00459		BRB	21\$		
06	09	AC	01	E1	0045B	18\$:	BBC	#1, QUEUE_STATUS+1, 19\$	2457	
			CF	9F	00460		PUSHAB	P.AFQ	2459	
			25	11	00464		BRB	21\$		
		F9BE	CF	9F	00466	19\$:	PUSHAB	P.AFR	2462	
	68		01	FB	0046A		CALLS	#1, FAO		
	01		CD	D1	0046D		CMPL	JOB_COPIES, #1	2463	
			55	1B	00472		BLEQU	24\$		

7E	FCAC	CD		01	C1	00474	ADDL3	#1, JOB_COPIES_DONE, -(SP)	2464
			F9B5	CF	9F	0047A	PUSHAB	P.AFS	
				00DD	31	0047E	BRW	35\$	
07	FC74	CD		02	E1	00481	BBC	#2, JOB_STATUS, 22\$	2469
			F9B6	CF	9F	00487	PUSHAB	P.AFT	2471
				00D9	31	0048B	BRW	38\$	
37	FC74	CD		04	E1	0048E	BBC	#4, JOB_STATUS, 25\$	2474
			F9B3	CF	9F	00494	PUSHAB	P.AFU	2477
		68		01	FB	00498	CALLS	#1, FAO	
			00BE	CE	B5	0049B	TSTW	REFUSAL_REASON_LENGTH	2478
				3B	13	0049F	BEQL	26\$	
			FCF4	CD	D5	004A1	TSTL	COMPLETED_BLOCKS	2481
				0F	13	004A5	BEQL	23\$	
			FCF4	CD	DD	004A7	PUSHL	COMPLETED_BLOCKS	2484
			F9A6	CF	9F	004AB	PUSHAB	P.AFV	
		68		02	FB	004AF	CALLS	#2, FAO	
			FCF4	CD	D4	004B2	CLRL	COMPLETED_BLOCKS	2485
		6A		00	FB	004B6	CALLS	#0, EOL	2487
			00C0	CE	9F	004B9	PUSHAB	REFUSAL_REASON	2488
		7E	00C2	CE	3C	004BD	MOVZWL	REFUSAL_REASON_LENGTH, -(SP)	
			F99D	CF	9F	004C2	PUSHAB	P.AFW	
		68		03	FB	004C6	CALLS	#3, FAO	
				78	11	004C9	BRB	32\$	2474
			FC74	CD	95	004CB	TSTB	JOB_STATUS	2493
				74	18	004CF	BGEQ	33\$	
			F996	CF	9F	004D1	PUSHAB	P.AFX	2496
		68		01	FB	004D5	CALLS	#1, FAO	
			FCE8	CD	D5	004D8	TSTL	CONDITION_VECTOR	2497
				03	12	004DC	BNEQ	27\$	
				0080	31	004DE	BRW	36\$	
		7B	FCE8	CD	E8	004E1	BLBS	CONDITION_VECTOR, 36\$	2498
				52	D4	004E6	CLRL	I	2501
			FCE8	CD	D5	004E8	TSTL	CONDITION_VECTOR[I]	2503
				50	13	004ED	BEQL	31\$	
	0084	CE	84	8F	9A	004EF	MOVZBL	#132, MESSAGE_DESC	2510
	0088	CE		6E	9E	004F5	MOVAB	MESSAGE, MESSAGE_DESC+4	2511
		7E		0F	7D	004FA	MOVQ	#15, -(SP)	2516
			008C	CE	9F	004FD	PUSHAB	MESSAGE_DESC	
			0090	CE	9F	00501	PUSHAB	MESSAGE_DESC	
			FCE8	CD	DD	00505	PUSHL	CONDITION_VECTOR[I]	
00000000G		00		05	FB	0050A	CALLS	#5, SYS\$GETMSG	
				52	D5	00511	TSTL	I	2517
				03	13	00513	BEQL	29\$	
		6E		2D	90	00515	MOVB	#45, MESSAGE	
			FCF4	CD	D5	00518	TSTL	COMPLETED_BLOCKS	2518
				0F	13	0051C	BEQL	30\$	
			FCF4	CD	DD	0051E	PUSHL	COMPLETED_BLOCKS	2521
			F951	CF	9F	00522	PUSHAB	P.AFY	
		68		02	FB	00526	CALLS	#2, FAO	
			FCF4	CD	D4	00529	CLRL	COMPLETED_BLOCKS	2522
		6A		00	FB	0052D	CALLS	#0, EOL	2524
			0088	CE	DD	00530	PUSHL	MESSAGE_DESC+4	2526
			0088	CE	DD	00534	PUSHL	MESSAGE_DESC	
			F949	CF	9F	00538	PUSHAB	P.AFZ	
		68		03	FB	0053C	CALLS	#3, FAO	
A5		52		02	F3	0053F	AOBLEQ	#2, I, 28\$	2501
				25	11	00543	BRB	39\$	2493

		06	FC75	CD	E9	00545	33\$:	BLBC	JOB STATUS+1, 34\$	2533
			F93F	CF	9F	0054A		PUSHAB	P.AGA	2535
				17	11	0054E		BRB	38\$	
0D	FC75	CD		01	E1	00550	34\$:	BBC	#1, JOB STATUS+1, 37\$	2538
			FE38	CD	9F	00556		PUSHAB	AFTER_TIME	2540
			F93A	CF	9F	0055A		PUSHAB	P.AGB	
		68		02	FB	0055E	35\$:	CALLS	#2, FAO	
				07	11	00561	36\$:	BRB	39\$	
			F947	CF	9F	00563	37\$:	PUSHAB	P.AGC	2544
		68		01	FB	00567	38\$:	CALLS	#1, FAO	
			FCF4	CD	D5	0056A	39\$:	TSTL	COMPLETED_BLOCKS	2547
				0B	13	0056E		BEQL	40\$	
			FCF4	CD	DD	00570		PUSHL	COMPLETED_BLOCKS	2549
			F940	CF	9F	00574		PUSHAB	P.AGD	
		68		02	FB	00578		CALLS	#2, FAO	
		6A		00	FB	0057B	40\$:	CALLS	#0, EOL	2552
	F9	A9		04	8A	0057E		BICB2	#4, FLAGS	2553
03	FC74	CD		03	E1	00582		BBC	#3, JOB_STATUS, 41\$	2556
				02B0	31	00588		BRW	80\$	
03	F9	A9		01	E0	0058B	41\$:	BBS	#1, FLAGS, 42\$	2559
				029B	31	00590		BRW	78\$	
	FD	A9		04	90	00593	42\$:	MOVB	#4, FLAGS+4	2562
		69		04	90	00597		MOVB	#4, FLAGS+7	2563
			00B0	CE	9F	0059A		PUSHAB	SUBMISSION_TIME	2566
			F924	CF	9F	0059E		PUSHAB	P.AGE	
		68		02	FB	005A2		CALLS	#2, FAO	
		69		01	90	005A5		MOVB	#1, FLAGS+7	2567
		28		55	E9	005A8		BLBC	R5, 46\$	2570
22	FCA8	CD		03	E1	005AB		BBC	#3, JOB_FLAGS, 46\$	2573
06	FCA8	CD		01	E1	005B1		BBC	#1, JOB_FLAGS, 43\$	2576
			F91B	CF	9F	005B7		PUSHAB	P.AGF	2577
				10	11	005BB		BRB	45\$	
06	FCA8	CD		02	E1	005BD	43\$:	BBC	#2, JOB_FLAGS, 44\$	2578
			F916	CF	9F	005C3		PUSHAB	P.AGG	2579
				04	11	005C7		BRB	45\$	
			F91B	CF	9F	005C9	44\$:	PUSHAB	P.AGH	2580
		68		01	FB	005CD	45\$:	CALLS	#1, FAO	
		69		01	90	005D0		MOVB	#1, FLAGS+7	2581
			FE28	CD	9F	005D3	46\$:	PUSHAB	CHARACTERISTICS	2586
	FA48	C7		01	FB	005D7		CALLS	#1, FORMAT_CHARACTERISTICS	
		64		54	E9	005DC		BLBC	R4, 52\$	2589
			FCFA	CD	B5	005DF		TSTW	CLI_LENGTH	2592
				13	13	005E3		BEQL	47\$	
			FCFC	CD	9F	005E5		PUSHAB	CLI	2595
		7E	FCFA	CD	3C	005E9		MOVZWL	CLI_LENGTH, -(SP)	
			F8FF	CF	9F	005EE		PUSHAB	P.AGI	
		68		03	FB	005F2		CALLS	#3, FAO	
		69		01	90	005F5		MOVB	#1, FLAGS+7	2596
		0D	FCA8	CD	E9	005F8	47\$:	BLBC	JOB_FLAGS, 48\$	2600
			FCE4	CD	DD	005FD		PUSHL	CPU_LIMIT	2604
			F8F5	CF	9F	00601		PUSHAB	P.AGJ	2603
				02	FB	00605		CALLS	#2, FORMAT_CPU_TIME	
0A	FAB8	C7		02	E0	0060A	48\$:	BBS	#2, JOB_FLAGS+T, 49\$	2607
	FCA9	CD	F8EA	CF	9F	00610		PUSHAB	P.AGK	2610
		68		01	FB	00614		CALLS	#1, FAO	
		69		01	90	00617		MOVB	#1, FLAGS+7	2611
0A	FCA9	CD		03	E1	0061A	49\$:	BBC	#3, JOB_FLAGS+1, 50\$	2615

			68	F8E0	CF	9F	00620		PUSHAB	P.AGL		2618
			69		01	FB	00624		CALLS	#1, FAO		
					01	90	00627		MOVB	#1, FLAGS+7		2619
				FB46	CD	B5	0062A	50\$:	TSTW	LOG_SPECIFICATION_LENGTH		2623
					03	12	0062E		BNEQ	51\$		
					00A5	31	00630		BRW	63\$		
				FB48	CD	9F	00633	51\$:	PUSHAB	LOG_SPECIFICATION		2627
			7E	FB46	CD	3C	00637		MOVZWL	LOG_SPECIFICATION_LENGTH, -(SP)		
				F8CB	CF	9F	0063C		PUSHAB	P.AGM		
					008F	31	00640		BRW	62\$		
22	FCA8	CD			06	E1	00643	52\$:	BBC	#6, JOB_FLAGS, 56\$		2633
06	FCA8	CD			04	E1	00649		BBC	#4, JOB_FLAGS, 53\$		2636
				F8C1	CF	9F	0064F		PUSHAB	P.AGN		2637
					10	11	00653		BRB	55\$		
06	FCA8	CD			05	E1	00655	53\$:	BBC	#5, JOB_FLAGS, 54\$		2638
				F8BB	CF	9F	0065B		PUSHAB	P.AGO		2639
					04	11	0065F		BRB	55\$		
				F8BF	CF	9F	00661	54\$:	PUSHAB	P.AGP		2640
			68		01	FB	00665	55\$:	CALLS	#1, FAO		
			69		01	90	00668		MOVB	#1, FLAGS+7		2641
16	FCAA	CD			04	E1	0066B	56\$:	BBC	#4, JOB_FLAGS+2, 59\$		2645
06	FCAA	CD			03	E1	00671		BBC	#3, JOB_FLAGS+2, 57\$		2648
				F8B1	CF	9F	00677		PUSHAB	P.AGQ		2649
					04	11	0067B		BRB	58\$		
				F8B1	CF	9F	0067D	57\$:	PUSHAB	P.AGR		2650
			68		01	FB	00681	58\$:	CALLS	#1, FAO		
			69		01	90	00684		MOVB	#1, FLAGS+7		2651
				FCC0	CD	9F	00687	59\$:	PUSHAB	FORM_NAME		2655
			7E	FCBE	CD	3C	0068B		MOVZWL	FORM_NAME_LENGTH, -(SP)		
				F8A6	CF	9F	00690		PUSHAB	P.AGS		
			68		03	FB	00694		CALLS	#3, FAO		
			69		01	90	00697		MOVB	#1, FLAGS+7		2656
			01	FCB0	CD	D1	0069A		CMPL	JOB_COPIES, #1		2659
					0E	1B	0069F		BLEQU	60\$		
				FCB0	CD	DD	006A1		PUSHL	JOB_COPIES		2662
				F89B	CF	9F	006A5		PUSHAB	P.AGT		
			68		02	FB	006A9		CALLS	#2, FAO		
			69		01	90	006AC		MOVB	#1, FLAGS+7		2663
0A	FCA9	CD			05	E1	006AF	60\$:	BBC	#5, JOB_FLAGS+1, 61\$		2667
				F89A	CF	9F	006B5		PUSHAB	P.AGU		2670
			68		01	FB	006B9		CALLS	#1, FAO		
			69		01	90	006BC		MOVB	#1, FLAGS+7		2671
				FA42	CD	B5	006BF	61\$:	TSTW	NOTE_LENGTH		2675
					13	13	006C3		BEQL	63\$		
				FA44	CD	9F	006C5		PUSHAB	NOTE		2678
			7E	FA42	CD	3C	006C9		MOVZWL	NOTE_LENGTH, -(SP)		
				F888	CF	9F	006CE		PUSHAB	P.AGV		
			68		03	FB	006D2	62\$:	CALLS	#3, FAO		
			69		01	90	006D5		MOVB	#1, FLAGS+7		2679
0A	FCA9	CD			06	E1	006D8	63\$:	BBC	#6, JOB_FLAGS+1, 64\$		2684
				F884	CF	9F	006DE		PUSHAB	P.AGW		2687
			68		01	FB	006E2		CALLS	#1, FAO		
			69		01	90	006E5		MOVB	#1, FLAGS+7		2688
			19		55	E9	006E8	64\$:	BLBC	R5, 65\$		2692
				F93E	CD	B5	006EB		TSTW	OPERATOR_REQUEST_LENGTH		2695
					13	13	006EF		BEQL	65\$		
				F940	CD	9F	006F1		PUSHAB	OPERATOR_REQUEST		2699

		7E	F93E	CD	3C	006F5	MOVZWL	OPERATOR_REQUEST_LENGTH, -(SP)	
			F870	CF	9F	006FA	PUSHAB	P.AGX	
		68		03	FB	006FE	CALLS	#3, FAO	
		69		01	90	00701	MOVB	#1, FLAGS+7	2700
			01C8	CE	9F	00704	PUSHAB	PARAMETER_8	2705
		7E	01CA	CE	3C	00708	MOVZWL	PARAMETER_8_LENGTH, -(SP)	2713
			02D4	CE	9F	0070D	PUSHAB	PARAMETER_7	2705
		7E	02D6	CE	3C	00711	MOVZWL	PARAMETER_7_LENGTH, -(SP)	2712
			03E0	CE	9F	00716	PUSHAB	PARAMETER_6	2705
		7E	03E2	CE	3C	0071A	MOVZWL	PARAMETER_6_LENGTH, -(SP)	2711
			04EC	CE	9F	0071F	PUSHAB	PARAMETER_5	2705
		7E	04EE	CE	3C	00723	MOVZWL	PARAMETER_5_LENGTH, -(SP)	2710
			05F8	CE	9F	00728	PUSHAB	PARAMETER_4	2705
		7E	05FA	CE	3C	0072C	MOVZWL	PARAMETER_4_LENGTH, -(SP)	2709
			0704	CE	9F	00731	PUSHAB	PARAMETER_3	2705
		7E	0706	CE	3C	00735	MOVZWL	PARAMETER_3_LENGTH, -(SP)	2708
			0810	CE	9F	0073A	PUSHAB	PARAMETER_2	2705
		7E	0812	CE	3C	0073E	MOVZWL	PARAMETER_2_LENGTH, -(SP)	2707
			F83C	CD	9F	00743	PUSHAB	PARAMETER_1	2705
		7E	F83A	CD	3C	00747	MOVZWL	PARAMETER_1_LENGTH, -(SP)	2706
	FD90	C7		10	FB	0074C	CALLS	#16, FORMAT_PARAMETERS	
		29		54	E9	00751	BLBC	R4, 67\$	2716
OA	FCA9	CD		04	E0	00754	BBS	#4, JOB_FLAGS+1, 66\$	2719
			F820	CF	9F	0075A	PUSHAB	P.AGY	2722
		68		01	FB	0075E	CALLS	#1, FAO	
		69		01	90	00761	MOVB	#1, FLAGS+7	2723
			FC4A	CD	B5	00764	TSTW	LOG_QUEUE_LENGTH	2727
				13	13	00768	BEQL	67\$	
		7E	FC4C	CD	9F	0076A	PUSHAB	LOG_QUEUE	2730
			FC4A	CD	3C	0076E	MOVZWL	LOG_QUEUE_LENGTH, -(SP)	
			F810	CF	9F	00773	PUSHAB	P.AGZ	
		68		03	FB	00777	CALLS	#3, FAO	
		69		01	90	0077A	MOVB	#1, FLAGS+7	2731
			01C0	CE	DD	0077D	PUSHL	PRIORITY	2736
			F80D	CF	9F	00781	PUSHAB	P.AHA	
		68		02	FB	00785	CALLS	#2, FAO	
		69		01	90	00788	MOVB	#1, FLAGS+7	2737
		5D		54	E9	0078B	BLBC	R4, 72\$	2740
			FCA9	CD	95	0078E	TSTB	JOB_FLAGS+1	2743
				0A	18	00792	BGEQ	68\$	
			F804	CF	9F	00794	PUSHAB	P.AHB	2746
		68		01	FB	00798	CALLS	#1, FAO	
		69		01	90	0079B	MOVB	#1, FLAGS+7	2747
			FD26	CD	B5	0079E	TSTW	CHECKPOINT_DATA_LENGTH	2751
				13	13	007A2	BEQL	69\$	
		7E	FD28	CD	9F	007A4	PUSHAB	CHECKPOINT_DATA	2755
			FD26	CD	3C	007A8	MOVZWL	CHECKPOINT_DATA_LENGTH, -(SP)	
			F7F4	CF	9F	007AD	PUSHAB	P.AHC	
		68		03	FB	007B1	CALLS	#3, FAO	
		69		01	90	007B4	MOVB	#1, FLAGS+7	2756
		0B	FCAA	CD	E9	007B7	BLBC	JOB_FLAGS+2, 70\$	2760
			009C	CE	DD	007BC	PUSHL	WSDEFAULT	2764
			F7F5	CF	9F	007C0	PUSHAB	P.AHD	2763
		67		02	FB	007C4	CALLS	#2, FORMAT WORKING SET	
OB	FCAA	CD		01	E1	007C7	BBC	#1, JOB_FLAGS+2, 7T\$	2767
			0098	CE	DD	007CD	PUSHL	WSXTENT	2771
			F7EE	CF	9F	007D1	PUSHAB	P.AHE	2770

41	FCAA	67	02	FB	007D5	CALLS	#2, FORMAT_WORKING_SET	:	
		CD	02	E1	007D8	BBC	#2, JOB_FLAGS+2, 77\$:	2774
			0094	CE	DD	PUSHL	WSQUOTA	:	2778
		F7E6	CF	9F	007E2	PUSHAB	P.AHF	:	2777
		67	02	FB	007E6	CALLS	#2, FORMAT_WORKING_SET	:	
			34	11	007E9	BRB	77\$:	2740
		FCA9	CD	95	007EB	TSTB	JOB_FLAGS+1	:	2782
			0A	19	007EF	BLSS	73\$:	
		F7DF	CF	9F	007F1	PUSHAB	P.AHG	:	2785
		68	01	FB	007F5	CALLS	#1, FAO	:	
		69	01	90	007F8	MOVB	#1, FLAGS+7	:	2786
1E	FCA9	CD	01	E1	007FB	BBC	#1, JOB_FLAGS+1, 77\$:	2790
			FCA8	CD	95	TSTB	JOB_FLAGS	:	2793
			06	18	00805	BGEQ	74\$:	
		F7D4	CF	9F	00807	PUSHAB	P.AHH	:	2794
			0F	11	0080B	BRB	76\$:	
		06	FCA9	CD	E9	BLBC	JOB_FLAGS+1, 75\$:	2795
			F7D2	CF	9F	PUSHAB	P.ARI	:	2796
			04	11	00816	BRB	76\$:	
		F7D9	CF	9F	00818	PUSHAB	P.AHJ	:	2797
		68	01	FB	0081C	CALLS	#1, FAO	:	
			69	94	0081F	CLRB	FLAGS+7	:	2802
			FD	A9	94	CLRB	FLAGS+4	:	2803
			00	FB	00824	CALLS	#0, CEOL	:	2804
04	0000V	CF	01	E0	00829	BBS	#1, FLAGS, 79\$:	2808
	F9	A9	A9	E9	0082E	BLBC	FLAGS, 80\$:	
		09	F9	A9	E9	MOVQ	QUEUE_FLAGS, -(SP)	:	2810
		7E	04	AC	7D	CALLS	#2, SHOW FILES	:	
	0000V	CF	02	FB	00836	BBC	#1, FLAGS, 81\$:	2814
04	F9	A9	01	E1	0083B	MOVB	#1, FLAGS+6	:	
	FF	A9	01	90	00840	BRW	4\$:	2387
			FAFE	31	00844	RET		:	2816
			04	00847				:	

; Routine Size: 2120 bytes, Routine Base: CODE + 1951

```

2835 2817 1 ROUTINE SHOW_FILES(QUEUE_FLAGS,QUEUE_STATUS): NOVALUE=
2836 2818 1
2837 2819 1 !++
2838 2820 1
2839 2821 1 FUNCTIONAL DESCRIPTION:
2840 2822 1 This routine displays the files contained in a job.
2841 2823 1
2842 2824 1 INPUT PARAMETERS:
2843 2825 1 NONE
2844 2826 1
2845 2827 1 IMPLICIT INPUTS:
2846 2828 1 NONE
2847 2829 1
2848 2830 1 OUTPUT PARAMETERS:
2849 2831 1 NONE
2850 2832 1
2851 2833 1 IMPLICIT OUTPUTS:
2852 2834 1 NONE
2853 2835 1
2854 2836 1 ROUTINE VALUE:
2855 2837 1 NONE
2856 2838 1
2857 2839 1 SIDE EFFECTS:
2858 2840 1 NONE
2859 2841 1
2860 2842 1 --
2861 2843 1
2862 2844 2 BEGIN
2863 2845 2 MAP
2864 2846 2 QUEUE_FLAGS: BBLOCK,
2865 2847 2 QUEUE_STATUS: BBLOCK;
2866 2848 2 LOCAL
2867 2849 2 ITMLST: BBLOCK[112],
2868 2850 2 FILE_COPIES: VOLATILE,
2869 2851 2 FILE_COPIES_DONE: VOLATILE,
2870 2852 2 FILE_FLAGS: VOLATILE BBLOCK[4],
2871 2853 2 FILE_SETUP_MODULES: VOLATILE VECTOR[255,BYTE],
2872 2854 2 FILE_SETUP_MODULES_LENGTH: VOLATILE WORD,
2873 2855 2 FILE_SPECIFICATION: VOLATILE VECTOR[255,BYTE],
2874 2856 2 FILE_SPECIFICATION_LENGTH: VOLATILE WORD,
2875 2857 2 FILE_STATUS: VOLATILE BBLOCK[4],
2876 2858 2 FIRST_PAGE: VOLATILE,
2877 2859 2 LAST_PAGE: VOLATILE,
2878 2860 2 STATUS,
2879 2861 2 IOSB: VECTOR[2];
2880 2862 2
2881 2863 2
2882 2864 2 ! Initialize the item list.
2883 2865 2
2884 2866 2 ITMLST[0,0,16,0] = 4;
2885 2867 2 ITMLST[2,0,16,0] = QUI$ SEARCH_FLAGS;
2886 2868 2 ITMLST[4,0,32,0] = UPLIT(QUI$M_SEARCH_WILDCARD);
2887 2869 2 ITMLST[8,0,32,0] = 0;
2888 2870 2
2889 2871 2 ITMLST[12,0,16,0] = %ALLOCATION(FILE_COPIES);
2890 2872 2 ITMLST[14,0,16,0] = QUI$ FILE_COPIES;
2891 2873 2 ITMLST[16,0,32,0] = FILE_COPIES;

```



```
2892 2874 2 ITMLST[20,0,32,0] = 0;
2893 2875 2
2894 2876 2 ITMLST[24,0,16,0] = %ALLOCATION(FILE_COPIES_DONE);
2895 2877 2 ITMLST[26,0,16,0] = QUI$ FILE_COPIES_DONE;
2896 2878 2 ITMLST[28,0,32,0] = FILE_COPIES_DONE;
2897 2879 2 ITMLST[32,0,32,0] = 0;
2898 2880 2
2899 2881 2 ITMLST[36,0,16,0] = %ALLOCATION(FILE_SPECIFICATION);
2900 2882 2 ITMLST[38,0,16,0] = QUI$ FILE_SPECIFICATION;
2901 2883 2 ITMLST[40,0,32,0] = FILE_SPECIFICATION;
2902 2884 2 ITMLST[44,0,32,0] = FILE_SPECIFICATION_LENGTH;
2903 2885 2
2904 2886 2 ITMLST[48,0,16,0] = %ALLOCATION(FILE_STATUS);
2905 2887 2 ITMLST[50,0,16,0] = QUI$ FILE_STATUS;
2906 2888 2 ITMLST[52,0,32,0] = FILE_STATUS;
2907 2889 2 ITMLST[56,0,32,0] = 0;
2908 2890 2
2909 2891 2 ITMLST[60,0,32,0] = 0;
2910 2892 2
2911 2893 2 IF .FLAGS[V_FULL]
2912 2894 2 THEN
2913 2895 2 BEGIN
2914 2896 2 ITMLST[60,0,16,0] = %ALLOCATION(FILE_FLAGS);
2915 2897 2 ITMLST[62,0,16,0] = QUI$ FILE_FLAGS;
2916 2898 2 ITMLST[64,0,32,0] = FILE_FLAGS;
2917 2899 2 ITMLST[68,0,32,0] = 0;
2918 2900 2
2919 2901 2 ITMLST[72,0,16,0] = %ALLOCATION(FILE_SETUP_MODULES);
2920 2902 2 ITMLST[74,0,16,0] = QUI$ FILE_SETUP_MODULES;
2921 2903 2 ITMLST[76,0,32,0] = FILE_SETUP_MODULES;
2922 2904 2 ITMLST[80,0,32,0] = FILE_SETUP_MODULES_LENGTH;
2923 2905 2
2924 2906 2 ITMLST[84,0,16,0] = %ALLOCATION(FIRST_PAGE);
2925 2907 2 ITMLST[86,0,16,0] = QUI$ FIRST_PAGE;
2926 2908 2 ITMLST[88,0,32,0] = FIRST_PAGE;
2927 2909 2 ITMLST[92,0,32,0] = 0;
2928 2910 2
2929 2911 2 ITMLST[96,0,16,0] = %ALLOCATION(LAST_PAGE);
2930 2912 2 ITMLST[98,0,16,0] = QUI$ LAST_PAGE;
2931 2913 2 ITMLST[100,0,32,0] = LAST_PAGE;
2932 2914 2 ITMLST[104,0,32,0] = 0;
2933 2915 2
2934 2916 2 ITMLST[108,0,32,0] = 0;
2935 2917 2 END;
2936 2918 2
2937 2919 2
2938 2920 2 ! Loop to get all files.
2939 2921 2
2940 2922 2 WHILE TRUE DO
2941 2923 2 BEGIN
2942 2924 2 STATUS = $GETQUIW(
2943 2925 2 FUNC=QUI$ DISPLAY_FILE,
2944 2926 2 ITMLST=ITMLST,
2945 2927 2 IOSB=IOSB);
2946 2928 2 IF .STATUS THEN STATUS = .IOSB;
2947 2929 2 IF NOT .STATUS THEN EXITLOOP;
2948 2930 2
```

P
P
P

```
2949      2931 3
2950      2932 3
2951      2933 3
2952      2934 3
2953      2935 3
2954      2936 3
2955      2937 3
2956      2938 3
2957      2939 3
2958      2940 3
2959      2941 3
2960      2942 3
2961      2943 4
2962      2944 4
2963      2945 4
2964      2946 4
2965      2947 4
2966      2948 5
2967      2949 5
2968      2950 5
2969      2951 5
2970      2952 5
2971      2953 5
2972      2954 5
2973      2955 5
2974      2956 4
2975      2957 4
2976      2958 4
2977      2959 3
2978      2960 3
2979      2961 4
2980      2962 4
2981      2963 4
2982      2964 3
2983      2965 3
2984      2966 3
2985      2967 3
2986      2968 3
2987      2969 3
2988      2970 3
2989      2971 4
2990      2972 4
2991      2973 4
2992      2974 4
2993      2975 5
2994      2976 5
2995      2977 5
2996      2978 6
2997      2979 6
2998      2980 6
2999      2981 6
3000      2982 6
3001      2983 5
3002      2984 5
3003      2985 5
3004      2986 5
3005      2987 5

      P
      P

      IF .FILE_STATUS[QUI$V_FILE_EXECUTING]
      THEN
      BEGIN
      IF .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
      THEN
      FAO_('executing')
      ELSE
      BEGIN
      IF .QUEUE_STATUS[QUI$V_QUEUE_SERVER]
      THEN FAO_('processing')
      ELSE FAO_('printing');
      IF .FILE_COPIES GTRU 1
      THEN FAO_('copy !UL',
      .FILE_COPIES_DONE + 1);
      FAO_('');
      END;
      FLAGS[B_PENDING_SPACES] = 1;
      END
      ELSE IF .FILE_STATUS[QUI$V_FILE_CHECKPOINTED]
      THEN
      BEGIN
      FAO_('checkpointed');
      FLAGS[B_PENDING_SPACES] = 1;
      END;

      FLAGS[V_TRUNCATE] = FALSE;

      IF .FLAGS[V_FULL]
      THEN
      BEGIN
      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
      THEN
      BEGIN
      IF .FILE_FLAGS[QUI$V_FILE_BURST_EXP]
      THEN
      BEGIN
      IF .FILE_FLAGS[QUI$V_FILE_BURST]
      THEN FAO_('BURST')
      ELSE FAO_('NOBURST');
      FLAGS[B_PENDING_SPACES] = 1;
      END;

      IF .FILE_COPIES GTRU 1
      THEN
```



```
3006      2988 6      BEGIN
3007      2989 6      FAO ('/COPIES=!UL', .FILE_COPIES);
3008      2990 6      FLAGS[B_PENDING_SPACES] = 1;
3009      2991 5      END;
3010      2992 4      END;
3011      2993 4
3012      2994 4
3013      2995 4      IF .FILE_FLAGS[QUI$V_FILE_DELETE]
3014      2996 4      THEN
3015      2997 5      BEGIN
3016      2998 5      FAO ('/DELETE');
3017      2999 5      FLAGS[B_PENDING_SPACES] = 1;
3018      3000 4      END;
3019      3001 4
3020      3002 4
3021      3003 4      IF NOT .QUEUE_FLAGS[QUI$V_QUEUE_BATCH]
3022      3004 4      THEN
3023      3005 5      BEGIN
3024      3006 5      IF .FILE_FLAGS[QUI$V_FILE_PAGINATE_EXP]
3025      3007 5      THEN
3026      3008 6      BEGIN
3027      3009 6      IF .FILE_FLAGS[QUI$V_FILE_PAGINATE]
3028      3010 6      THEN FAO ('/FEED')
3029      3011 6      ELSE FAO ('/NOFEED');
3030      3012 6      FLAGS[B_PENDING_SPACES] = 1;
3031      3013 5      END;
3032      3014 5
3033      3015 5
3034      3016 5      IF .FILE_FLAGS[QUI$V_FILE_FLAG_EXP]
3035      3017 5      THEN
3036      3018 6      BEGIN
3037      3019 6      IF .FILE_FLAGS[QUI$V_FILE_FLAG]
3038      3020 6      THEN FAO ('/FLAG')
3039      3021 6      ELSE FAO ('/NOFLAG');
3040      3022 6      FLAGS[B_PENDING_SPACES] = 1;
3041      3023 5      END;
3042      3024 5
3043      3025 5
3044      3026 5      IF .FILE_FLAGS[QUI$V_FILE_PAGE_HEADER]
3045      3027 5      THEN
3046      3028 6      BEGIN
3047      3029 6      FAO ('/HEADER');
3048      3030 6      FLAGS[B_PENDING_SPACES] = 1;
3049      3031 5      END;
3050      3032 5
3051      3033 5
3052      3034 5      IF .FIRST_PAGE NEQ 0
3053      3035 5      THEN
3054      3036 5      IF .LAST_PAGE NEQ 0
3055      3037 5      THEN
3056      3038 6      BEGIN
3057      3039 6      FAO ('/PAGES=(!UL,!UL)', .FIRST_PAGE, .LAST_PAGE);
3058      3040 6      FLAGS[B_PENDING_SPACES] = 1;
3059      3041 6      END
3060      3042 5      ELSE
3061      3043 6      BEGIN
3062      3044 6      FAO ('/PAGES=(!UL,'')', .FIRST_PAGE);
```

```
3063      3045      6      FLAGS[B_PENDING_SPACES] = 1;
3064      3046      6      END
3065      3047      3      ELSE
3066      3048      3      IF .LAST_PAGE NEQ 0
3067      3049      3      THEN
3068      3050      6      BEGIN
3069      3051      6      FAO ('/PAGES=!UL', .LAST_PAGE);
3070      3052      6      FLAGS[B_PENDING_SPACES] = 1;
3071      3053      3      END;
3072      3054      3
3073      3055      3
3074      3056      3      IF .FILE_FLAGS[QUI$V_FILE_PASSALL]
3075      3057      3      THEN
3076      3058      6      BEGIN
3077      3059      6      FAO ('/PASSALL');
3078      3060      6      FLAGS[B_PENDING_SPACES] = 1;
3079      3061      3      END;
3080      3062      3
3081      3063      3
3082      3064      3      IF .FILE_SETUP_MODULES_LENGTH NEQ 0
3083      3065      3      THEN
3084      3066      6      BEGIN
3085      3067      6      P FAO ('/SETUP=(!AF)',
3086      3068      6      .FILE_SETUP_MODULES_LENGTH, FILE_SETUP_MODULES);
3087      3069      6      FLAGS[B_PENDING_SPACES] = 1;
3088      3070      3      END;
3089      3071      3
3090      3072      3
3091      3073      3      IF .FILE_FLAGS[QUI$V_FILE_DOUBLE_SPACE]
3092      3074      3      THEN
3093      3075      6      BEGIN
3094      3076      6      FAO ('/SPACE');
3095      3077      6      FLAGS[B_PENDING_SPACES] = 1;
3096      3078      3      END;
3097      3079      3
3098      3080      3
3099      3081      3      IF .FILE_FLAGS[QUI$V_FILE_TRAILER_EXP]
3100      3082      3      THEN
3101      3083      6      BEGIN
3102      3084      6      IF .FILE_FLAGS[QUI$V_FILE_TRAILER]
3103      3085      6      THEN FAO ('/TRAILER')
3104      3086      6      ELSE FAO ('/NOTRAILER');
3105      3087      6      FLAGS[B_PENDING_SPACES] = 1;
3106      3088      3      END;
3107      3089      3      END;
3108      3090      3      END;
3109      3091      3
3110      3092      3
3111      3093      3      FLAGS[B_LEFT_MARGIN] = 0;
3112      3094      3      FLAGS[B_PENDING_SPACES] = 0;
3113      3095      3      EOL ();
3114      3096      3      END;
3115      3097      1      END;
```



```
00000002 0219C P.ANK: .LONG 2
29 67 6E 69 74 75 63 65 78 65 21 07 021A0 P.AHL: .ASCII <7>\!6* !AF\
67 6E 69 73 73 65 63 6F 72 70 28 0B 021A8 P.AHM: .ASCII <11>\(executing)\
67 6E 69 74 6E 69 72 70 28 0B 021B4 P.AHN: .ASCII <11>\(processing)\
4C 55 21 20 79 70 6F 63 20 09 021C0 P.AHO: .ASCII <9>\(printing)\
29 64 65 74 6E 69 6F 70 68 63 65 68 63 29 01 021CA P.AHP: .ASCII <9>\ copy !UL\
54 53 52 55 42 55 42 2F 06 021D4 P.AHQ: .ASCII <1>\)\
53 45 49 50 4F 43 2F 0B 021D6 P.AHR: .ASCII <14>\(checkpointed)\
45 54 45 4C 45 44 2F 07 021E5 P.AHS: .ASCII <6>\BURST\
44 45 45 46 4F 4E 2F 08 021EC P.AHT: .ASCII <8>\NOBURST\
47 41 4C 46 4F 4E 2F 07 021F5 P.AHU: .ASCII <11>\COPIES=!UL\
52 45 44 41 45 48 2F 07 02201 P.AHV: .ASCII <7>\DELETE\
22 22 2C 4C 55 21 28 3D 53 45 47 41 50 2F 05 02209 P.AHW: .ASCII <5>\FEED\
4C 55 21 3D 53 45 47 41 50 2F 07 0220F P.AHX: .ASCII <7>\NOFEED\
29 46 41 21 28 3D 50 55 54 45 53 2F 05 02217 P.AHY: .ASCII <5>\FLAG\
52 45 4C 49 41 52 54 4F 4E 2F 07 0221D P.AHZ: .ASCII <7>\NOFLAG\
52 45 4C 49 41 52 54 4F 4E 2F 07 02225 P.AIA: .ASCII <7>\HEADER\
55 21 2C 4C 55 21 28 3D 53 45 47 41 50 2F 10 0222D P.AIB: .ASCII <16>\PAGES=(!UL,!UL)\
22 22 2C 4C 55 21 28 3D 53 45 47 41 50 2F 0F 0223C P.AIC: .ASCII <15>\PAGES=(!UL,'')\
4C 55 21 3D 53 45 47 41 50 2F 0A 0224E P.AID: .ASCII <10>\PAGES=!UL\
29 46 41 21 28 3D 50 55 54 45 53 2F 08 02259 P.AIE: .ASCII <8>\PASSALL\
52 45 4C 49 41 52 54 4F 4E 2F 0C 02262 P.AIF: .ASCII <12>\SETUP=(!AF)\
52 45 4C 49 41 52 54 4F 4E 2F 06 0226F P.AIG: .ASCII <6>\SPACE\
52 45 4C 49 41 52 54 4F 4E 2F 08 02276 P.AIH: .ASCII <8>\TRAILER\
52 45 4C 49 41 52 54 4F 4E 2F 0A 0227F P.AII: .ASCII <10>\NOTRAILER\
```

003C 00000 SHOW_FILES:

```
55 0000V CF 9E 00002 .WORD Save R2,R3,R4,R5 2817
54 0000' CF 9E 00007 MOVAB FA0, R5
53 FF02 CF 9E 0000C MOVAB FLAGS+7, R4
5E FD68 CE 9E 00011 MOVAB P.AHK, R3
90 AD 004C0004 8F D0 00016 MOVAB -664(SP), SP
94 AD 63 9E 0001E MOVAB #4980740, ITMLST
98 AD D4 00022 MOVAB P.AHK, ITMLST+4
9C AD 00100004 8F D0 00025 CLRL ITMLST+8
A0 AD 8C AD 9E 0002D MOVAB #1048580, ITMLST+12
A4 AD D4 00032 MOVAB FILE_COPIES, ITMLST+16
A8 AD 00120004 8F D0 00035 CLRL ITMLST+20
AC AD 88 AD 9E 0003D MOVAB #1179652, ITMLST+24
B0 AD D4 00042 MOVAB FILE_COPIES_DONE, ITMLST+28
B4 AD 001500FF 8F D0 00045 CLRL ITMLST+32
B8 AD 18 AE 9E 0004D MOVAB #1376511, ITMLST+36
BC AD 16 AE 9E 00052 MOVAB FILE_SPECIFICATION, ITMLST+40
C0 AD 00160004 8F D0 00057 MOVAB FILE_SPECIFICATION_LENGTH, ITMLST+44
C4 AD 10 AE 9E 0005F MOVAB #144T796, ITMLST+48
C8 AD 7C 00064 MOVAB FILE_STATUS, ITMLST+52
44 F9 A4 01 E1 00067 CLRL ITMLST+56
CC AD 00130004 8F D0 0006C BBC #1, FLAGS, 1$
D0 AD 84 AD 9E 00074 MOVAB #1245188, ITMLST+60
D8 AD 001400FF 8F D0 00079 MOVAB FILE_FLAGS, ITMLST+64
DC AD 011C CE 9E 00084 CLRL ITMLST+68
MOVAB #1310975, ITMLST+72
MOVAB FILE_SETUP_MODULES, ITMLST+76 2903
```

	E0	AD	011A	CE	9E	0008A	MOVAB	FILE SETUP MODULES LENGTH, ITMLST+80	2904
	E4	AD	00170004	8F	D0	00090	MOVL	#1507332, ITMLST+84	2906
	E8	AD	0C	AE	9E	00098	MOVAB	FIRST PAGE, ITMLST+88	2908
			EC	AD	D4	0009D	CLRL	ITMLST+92	2909
	F0	AD	00320004	8F	D0	000A0	MOVL	#3276804, ITMLST+96	2911
	F4	AD	08	AE	9E	000A8	MOVAB	LAST PAGE, ITMLST+100	2913
			F8	AD	7C	000AD	CLRL	ITMLST+104	2914
				7E	7C	000B0	CLRL	-(SP)	2927
			08	AE	9F	000B2	PUSHAB	IOSB	
			90	AD	9F	000B5	PUSHAB	ITMLST	
		7E		03	7D	000B8	MOVQ	#3, -(SP)	
				7E	D4	000BB	CLRL	-(SP)	
00000000G		00		07	FB	000BD	CALLS	#7, SYSSGETQUIW	
		52		50	D0	000C4	MOVL	R0, STATUS	
		03		52	E9	000C7	BLBC	STATUS, 2\$	2928
		52		6E	D0	000CA	MOVL	IOSB, STATUS	
		01		52	E8	000CD	BLBS	STATUS, 3\$	2929
					04	000D0	RET		
	F9	A4		04	88	000D1	BISB2	#4, FLAGS	2932
			18	AE	9F	000D5	PUSHAB	FILE_SPECIFICATION	2934
		7E	1A	AE	3C	000D8	MOVZWL	FILE_SPECIFICATION_LENGTH, -(SP)	
			04	A3	9F	000DC	PUSHAB	P.AHC	
		65		03	FB	000DF	CALLS	#3, FAO	
		64		01	90	000E2	MOVB	#1, FLAGS+7	2937
	FD	A4		08	90	000E5	MOVB	#8, FLAGS+4	2938
2F	10	AE		01	E1	000E9	BBC	#1, FILE STATUS, 8\$	2941
		05		04	AC	E9	BLBC	QUEUE_FLAGS, 4\$	2944
			0C	A3	9F	000F2	PUSHAB	P.AHM	2946
				2D	11	000F5	BRB	9\$	
05	09	AC		01	E1	000F7	BBC	#1, QUEUE_STATUS+1, 5\$	2949
			18	A3	9F	000FC	PUSHAB	P.AHN	2950
				03	11	000FF	BRB	6\$	
			24	A3	9F	00101	PUSHAB	P.AHO	2951
		65		01	FB	00104	CALLS	#1, FAO	
		01	8C	AD	D1	00107	CMPL	FILE_COPIES, #1	2952
				0B	1B	0010B	BLEQU	7\$	
7E	88	AD		01	C1	0010D	ADDL3	#1, FILE_COPIES_DONE, -(SP)	2954
			2E	A3	9F	00112	PUSHAB	P.AHP	
		65		02	FB	00115	CALLS	#2, FAO	
			38	A3	9F	00118	PUSHAB	P.AHQ	2955
				07	11	0011B	BRB	9\$	
		09	10	AE	E9	0011D	BLBC	FILE STATUS, 10\$	2959
			3A	A3	9F	00121	PUSHAB	P.AHR	2962
		65		01	FB	00124	CALLS	#1, FAO	
		64		01	90	00127	MOVB	#1, FLAGS+7	2963
	F9	A4		04	8A	0012A	BICB2	#4, FLAGS	2967
03	F9	A4		01	E0	0012E	BBS	#1, FLAGS, 12\$	2970
			0103	31	00133	BRW	35\$		
		29	04	AC	E8	00136	BLBS	QUEUE_FLAGS, 16\$	2973
				01	E1	0013A	BBC	#1, FILE_FLAGS, 15\$	2976
12	84	AD		AD	E9	0013F	BLBC	FILE_FLAGS, 13\$	2979
		05		A3	9F	00143	PUSHAB	P.AHS	2980
			84	03	11	00146	BRB	14\$	
			50	A3	9F	00148	PUSHAB	P.AHT	2981
		65		01	FB	0014B	CALLS	#1, FAO	
		64		01	90	0014E	MOVB	#1, FLAGS+7	2982
		01	8C	AD	D1	00151	CMPL	FILE_COPIES, #1	2986

			8C	0C	1B	00155	BLEQU	16\$		
			59	AD	DD	00157	PUSHL	FILE_COPIES		2989
		65		A3	9F	0015A	PUSHAB	P.AHO		
		64		02	FB	0015D	CALLS	#2, FAO		
09	84	AD		01	90	00160	MOVB	#1, FLAGS+7		2990
			65	02	E1	00163	BBC	#2, FILE_FLAGS, 17\$		2995
				A3	9F	00168	PUSHAB	P.AHV		2998
		65		01	FB	0016B	CALLS	#1, FAO		
		64		01	90	0016E	MOVB	#1, FLAGS+7		2999
13	85	BE	04	AC	E8	00171	BLBS	QUEUE_FLAGS, 11\$		3003
05	85	AD		03	E1	00175	BBC	#3, FILE_FLAGS+1, 20\$		3006
		AD		01	E1	0017A	BBC	#1, FILE_FLAGS+1, 18\$		3009
			6D	A3	9F	0017F	PUSHAB	P.AHW		3010
				03	11	00182	BRB	19\$		
			73	A3	9F	00184	PUSHAB	P.AHX		3011
		65		01	FB	00187	CALLS	#1, FAO		
		64		01	90	0018A	MOVB	#1, FLAGS+7		3012
14	84	AD		05	E1	0018D	BBC	#5, FILE_FLAGS, 23\$		3016
05	84	AD		04	E1	00192	BBC	#4, FILE_FLAGS, 21\$		3019
			7B	A3	9F	00197	PUSHAB	P.AHY		3020
				04	11	0019A	BRB	22\$		
			0081	C3	9F	0019C	PUSHAB	P.AHZ		3021
		65		01	FB	001A0	CALLS	#1, FAO		
		64		01	90	001A3	MOVB	#1, FLAGS+7		3022
		0A	85	AD	E9	001A6	BLBC	FILE_FLAGS+1, 24\$		3026
			0089	C3	9F	001AA	PUSHAB	P.AIX		3029
		65		01	FB	001AE	CALLS	#1, FAO		
		64		01	90	001B1	MOVB	#1, FLAGS+7		3030
			0C	AE	D5	001B4	TSTL	FIRST_PAGE		3034
				1D	13	001B7	BEQL	26\$		
			08	AE	D5	001B9	TSTL	LAST_PAGE		3036
				0F	13	001BC	BEQL	25\$		
			08	AE	DD	001BE	PUSHL	LAST_PAGE		3039
			10	AE	DD	001C1	PUSHL	FIRST_PAGE		
			0091	C3	9F	001C4	PUSHAB	P.AIB		
		65		03	FB	001C8	CALLS	#3, FAO		3036
				18	11	001CB	BRB	28\$		
			0C	AE	DD	001CD	PUSHL	FIRST_PAGE		3044
			00A2	C3	9F	001D0	PUSHAB	P.AIC		
				0C	11	001D4	BRB	27\$		
			08	AE	D5	001D6	TSTL	LAST_PAGE		3048
				0D	13	001D9	BEQL	29\$		
			08	AE	DD	001DB	PUSHL	LAST_PAGE		3051
			00B2	C3	9F	001DE	PUSHAB	P.AID		
		65		02	FB	001E2	CALLS	#2, FAO		
		64		01	90	001E5	MOVB	#1, FLAGS+7		3052
0A	85	AD		02	E1	001E8	BBC	#2, FILE_FLAGS+1, 30\$		3056
			00BD	C3	9F	001ED	PUSHAB	P.AIE		3059
		65		01	FB	001F1	CALLS	#1, FAO		
		64		01	90	001F4	MOVB	#1, FLAGS+7		3060
			011A	CE	B5	001F7	TSTW	FILE_SETUP_MODULES_LENGTH		3064
				13	13	001FB	BEQL	31\$		
			011C	CE	9F	001FD	PUSHAB	FILE_SETUP_MODULES		3068
		7E	011E	CE	3C	00201	MOVZWL	FILE_SETUP_MODULES_LENGTH, -(SP)		
			00C6	C3	9F	00206	PUSHAB	P.AIF		
		65		03	FB	0020A	CALLS	#3, FAO		
		64		01	90	0020D	MOVB	#1, FLAGS+7		3069

QUEMANSHO
V04-000

SHOW QUEUE command

M 8
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 103
(19)

0A	84	AD	00D3	03	E1	00210	31\$:	BBC	#3, FILE_FLAGS, 32\$:	3073
				C3	9F	00215		PUSHAB	P.AIG	:	3076
	65			01	FB	00219		CALLS	#1, FAO	:	
	64			01	90	0021C		MOVB	#1, FLAGS+7	:	3077
		84		AD	95	0021F	32\$:	TSTB	FILE_FLAGS	:	3081
				15	18	00222		BGEQ	35\$:	
06	84	AD	00DA	06	E1	00224		BBC	#6, FILE_FLAGS, 33\$:	3084
				C3	9F	00229		PUSHAB	P.AIH	:	3085
				04	11	0022D		BRB	34\$:	
			00E3	C3	9F	0022F	33\$:	PUSHAB	P.AII	:	3086
	65			01	FB	00233	34\$:	CALLS	#1, FAO	:	
	64			01	90	00236		MOVB	#1, FLAGS+7	:	3087
		FD		A4	94	00239	35\$:	CLRB	FLAGS+4	:	3093
				64	94	0023C		CLRB	FLAGS+7	:	3094
0000V	CF			00	FB	0023E		CALLS	#0, EOL	:	3095
			FE6A	31	00243			BRW	1\$:	2922
				04	00246			RET		:	3097

; Routine Size: 583 bytes, Routine Base: CODE + 228A


```
3117 3098 1 ROUTINE FAO(CTRL,PARAM): NOVALUE=
3118 3099 1
3119 3100 1 !++
3120 3101 1
3121 3102 1 FUNCTIONAL DESCRIPTION:
3122 3103 1 This routine interfaces to FAO to format information into the line.
3123 3104 1 Deferred blank lines and spaces are written before the requested
3124 3105 1 information, and unless truncation is in effect, if the requested
3125 3106 1 information does not fit on the current line, the information is placed
3126 3107 1 on the next line.
3127 3108 1
3128 3109 1 INPUT PARAMETERS:
3129 3110 1 CTRL - ASCII control string
3130 3111 1 PARAM... - Parameters required by the control string (if any)
3131 3112 1
3132 3113 1 IMPLICIT INPUTS:
3133 3114 1 NONE
3134 3115 1
3135 3116 1 OUTPUT PARAMETERS:
3136 3117 1 NONE
3137 3118 1
3138 3119 1 IMPLICIT OUTPUTS:
3139 3120 1 Information formatted into the line buffer.
3140 3121 1
3141 3122 1 ROUTINE VALUE:
3142 3123 1 NONE
3143 3124 1
3144 3125 1 SIDE EFFECTS:
3145 3126 1 NONE
3146 3127 1
3147 3128 1 --
3148 3129 1
3149 3130 2 BEGIN
3150 3131 2 MAP
3151 3132 2 CTRL: REF VECTOR[.BYTE];
3152 3133 2 LOCAL
3153 3134 2 OUTLEN: WORD,
3154 3135 2 DESC: VECTOR[2],
3155 3136 2 STATUS;
3156 3137 2
3157 3138 2
3158 3139 2 WHILE .FLAGS[B_PENDING_LINES] NEQ 0 DO
3159 3140 2 BEGIN
3160 3141 2 EOL();
3161 3142 2 FLAGS[B_PENDING_LINES] = .FLAGS[B_PENDING_LINES] - 1;
3162 3143 2 END;
3163 3144 2
3164 3145 2
3165 3146 2 DESC[0] = .CTRL[0];
3166 3147 2 DESC[1] = CTRL[1];
3167 3148 2
3168 3149 2
3169 3150 2 IF
3170 3151 2 BEGIN
3171 3152 2 IF .FLAGS[B_PENDING_SPACES] GTRU .LIST_DESC[0]
3172 3153 2 THEN
3173 3154 2 TRUE
```

```
3174 ELSE
3175 BEGIN
3176 IF .FLAGS[B_PENDING_SPACES] NEQ 0
3177 THEN
3178 BEGIN
3179 LIST_DESC[0] = .LIST_DESC[0] - .FLAGS[B_PENDING_SPACES];
3180 LIST_DESC[1] = CH$FICL(%C', .FLAGS[B_PENDING_SPACES], .LIST_DESC[1]);
3181 END;
3182
3183 STATUS = $FAOL(
3184 CTRSTR=DESC,
3185 OUTLEN=OUTLEN,
3186 OUTBUF=LIST_DESC,
3187 PRMLST=PARAM);
3188 .STATUS<0,16> EQL $$$_BUFFEROVF AND NOT .FLAGS[V_TRUNCATE]
3189 END
3190 THEN
3191 BEGIN
3192 EOL();
3193 LIST_DESC[0] = .LIST_DESC[0] - .FLAGS[B_LEFT_MARGIN];
3194 LIST_DESC[1] = CH$FICL(%C', .FLAGS[B_LEFT_MARGIN], .LIST_DESC[1]);
3195 $FAOL(CTRSTR=DESC, OUTLEN=OUTLEN, OUTBUF=LIST_DESC, PRMLST=PARAM);
3196 END;
3197
3198 FLAGS[B_PENDING_SPACES] = 0;
3199 LIST_DESC[0] = .LIST_DESC[0] - .OUTLEN;
3200 LIST_DESC[1] = .LIST_DESC[1] + .OUTLEN;
3201 END;
```

				.EXTRN	SYSSFAOL		
			00FC 00000	FAO:	.WORD	Save R2,R3,R4,R5,R6,R7	3098
		57 000C0000G	00 9E 00002		MOVAB	SYSSFAOL, R7	
		56 0000'	CF 9E 00009		MOVAB	LIST_DESC, R6	
		5E	0C C2 0000E		SUBL2	#12, SP	
			FE A6 95 00011	1\$:	TSTB	FLAGS+6	3139
			0A 13 00014		BEQL	2\$	
		0000V CF	00 FB 00016		CALLS	#0, EOL	3141
			FE A6 97 0001B		DECB	FLAGS+6	3142
			F1 11 0001E		BRB	1\$	3139
		04 AE	04 BC 9A 00020	2\$:	MOVZBL	@CTRL, DESC	3146
		04 AC	01 C1 00025		ADDL3	#1, CTRL, DESC+4	3147
		50	A6 9A 0002B		MOVZBL	FLAGS+7, R0	3152
		66	50 D1 0002F		CMPL	R0, LIST_DESC	
			2C 1A 00032		BGTRU	4\$	
			50 D5 00034		TSTL	R0	3157
			0E 13 00036		BEQL	3\$	
		66	50 C2 00038		SUBL2	R0, LIST_DESC	3160
50	20	6E	00 2C 0003B		MOVCS	#0, (SP), #32, R0, @LIST_DESC+4	3161
			04 B6				
		04 A6	53 D0 00042		MOVL	R3, LIST_DESC+4	
			08 AC 9F 00046	3\$:	PUSHAB	PARAM	3169

			08	56	DD	00049	PUSHL	R6		
			10	AE	9F	0004B	PUSHAB	OUTLEN		
				AE	9F	0004E	PUSHAB	DESC		
	0601	67		04	FB	00051	CALLS	#4, SYSSFAOL		
		8F		50	B1	00054	CMPW	STATUS, #1537		3170
				2E	12	00059	BNEQ	5\$		
29	F8	A6		02	E0	0005B	BBS	#2, FLAGS, 5\$		
	0000V	CF		00	FB	00060	CALLS	#0, EOL		3175
		50	FC	A6	9A	00065	MOVZBL	FLAGS+4, R0		3176
		66		50	C2	00069	SUBL2	R0, LIST_DESC		
		50	FC	A6	9A	0006C	MOVZBL	FLAGS+4, R0		3177
50		6E		00	2C	00070	MOVCS	#0, (SP), #32, R0, @LIST_DESC+4		
			04	B6		00075				
	04	A6		53	DO	00077	MOVL	R3, LIST_DESC+4		
			08	AC	9F	0007B	PUSHAB	PARAM		3178
				56	DD	0007E	PUSHL	R6		
			08	AE	9F	00080	PUSHAB	OUTLEN		
		67	10	AE	9F	00083	PUSHAB	DESC		
				04	FB	00086	CALLS	#4, SYSSFAOL		3182
		50	FF	A6	94	00089	CLRB	FLAGS+7		3183
		66		6E	3C	0008C	MOVZWL	OUTLEN, R0		
		50		50	C2	0008F	SUBL2	R0, LIST_DESC		3184
		50		6E	3C	00092	MOVZWL	OUTLEN, R0		
	04	A6		50	C0	00095	ADDL2	R0, LIST_DESC+4		3185
				04		00099	RET			

; Routine Size: 154 bytes, Routine Base: CODE + 24D1

```
3206 3186 1 ROUTINE EOL: NOVALUE=
3207 3187 1
3208 3188 1 !++
3209 3189 1
3210 3190 1 FUNCTIONAL DESCRIPTION:
3211 3191 1 This routine writes the listing buffer to the listing file.
3212 3192 1
3213 3193 1 INPUT PARAMETERS:
3214 3194 1 NONE
3215 3195 1
3216 3196 1 IMPLICIT INPUTS:
3217 3197 1 NONE
3218 3198 1
3219 3199 1 OUTPUT PARAMETERS:
3220 3200 1 NONE
3221 3201 1
3222 3202 1 IMPLICIT OUTPUTS:
3223 3203 1 NONE
3224 3204 1
3225 3205 1 ROUTINE VALUE:
3226 3206 1 NONE
3227 3207 1
3228 3208 1 SIDE EFFECTS:
3229 3209 1 The listing is produced.
3230 3210 1
3231 3211 1 !--
3232 3212 1
3233 3213 2 BEGIN
3234 3214 2 LOCAL
3235 3215 2 DESC: VECTOR[2]; ! Descriptor for line
3236 3216 2
3237 3217 2
3238 3218 2 DESC[0] = .FLAGS[B_RIGHT_MARGIN] - .LIST_DESC[0];
3239 3219 2 DESC[1] = LIST_BUFFER;
3240 3220 2 PUT_OUTPUT(DESC);
3241 3221 2 LIST_DESC[0] = .FLAGS[B_RIGHT_MARGIN];
3242 3222 2 LIST_DESC[1] = LIST_BUFFER;
3243 3223 2 CH$FILL(%' ', .LIST_DESC[0], .LIST_DESC[1]);
3244 3224 1 END;
```

				007C 00000	EOL:	.WORD	Save R2,R3,R4,R5,R6	3186
	56	0000'	CF	9E	00002	MOVAB	LIST_DESC, R6	
	5E		04	C2	00007	SUBL2	#4, SP	
	50		FD	A6	9A	MOVZBL	FLAGS+5, R0	3218
7E	50			66	C3	SUBL3	LIST_DESC, R0, DESC	
	04	AE	08	A6	9E	MOVAB	LIST_BUFFER, DESC+4	3219
				5E	DD	PUSHL	SP	3220
	0000V	CF		01	FB	CALLS	#1, PUT_OUTPUT	
		66	FD	A6	9A	MOVZBL	FLAGS+5, LIST_DESC	3221
		04	08	A6	9E	MOVAB	LIST_BUFFER, LIST_DESC+4	3222
66	20	6E		00	2C	MOVCS	#0, TSP), #32, LIST_DESC, @LIST_DESC+4	3223
			04	B6				
				04	0002E	RET		3224

QUEMANSHO
V04-000

SHOW QUEUE command

E 9
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 108
(21)

; Routine Size: 47 bytes, Routine Base: CODE + 256B

```

: 3246      3225 1 ROUTINE CEOL: NOVALUE=
: 3247      3226 1
: 3248      3227 1 ++
: 3249      3228 1
: 3250      3229 1 FUNCTIONAL DESCRIPTION:
: 3251      3230 1     This routine writes the listing buffer to the listing file if the
: 3252      3231 1     listing buffer contains any text.
: 3253      3232 1
: 3254      3233 1 INPUT PARAMETERS:
: 3255      3234 1     NONE
: 3256      3235 1
: 3257      3236 1 IMPLICIT INPUTS:
: 3258      3237 1     NONE
: 3259      3238 1
: 3260      3239 1 OUTPUT PARAMETERS:
: 3261      3240 1     NONE
: 3262      3241 1
: 3263      3242 1 IMPLICIT OUTPUTS:
: 3264      3243 1     NONE
: 3265      3244 1
: 3266      3245 1 ROUTINE VALUE:
: 3267      3246 1     NONE
: 3268      3247 1
: 3269      3248 1 SIDE EFFECTS:
: 3270      3249 1     The listing is produced.
: 3271      3250 1
: 3272      3251 1 --
: 3273      3252 1
: 3274      3253 2 BEGIN
: 3275      3254 2 IF .LIST_DESC[0] NEQ .FLAGS[B_RIGHT_MARGIN] THEN EOL();
: 3276      3255 1 END;

```

0000' CF	0000' CF	08	0000 00000 CEOL:	.WORD	Save nothing	: 3225
			00 ED 00002	CMPZV	#0, #8, FLAGS+5, LIST_DESC	: 3254
			04 13 0000B	BEQL	1\$:
			00 FB 0000D	CALLS	#0, EOL	:
			04 00011 1\$:	RET		: 3255

; Routine Size: 18 bytes, Routine Base: CODE + 259A


```
3278 1 ROUTINE PUT_OUTPUT (DESC) : NOVALUE=
3279 1
3280 1 ++
3281 1
3282 1 FUNCTIONAL DESCRIPTION:
3283 1 This routine writes a record to the listing file.
3284 1
3285 1 INPUT PARAMETERS:
3286 1 DESC = The address of the string descriptor for the record to write.
3287 1
3288 1 IMPLICIT INPUTS:
3289 1 NONE
3290 1
3291 1 OUTPUT PARAMETERS:
3292 1 NONE
3293 1
3294 1 IMPLICIT OUTPUTS:
3295 1 NONE
3296 1
3297 1 ROUTINE VALUE:
3298 1 NONE
3299 1
3300 1 SIDE EFFECTS:
3301 1 The listing is produced.
3302 1
3303 1 --
3304 1
3305 2 BEGIN
3306 2 MAP
3307 2 DESC: REF BBLOCK;
3308 2
3309 2 LOCAL
3310 2 STATUS;
3311 2
3312 2
3313 2 RAB[RAB$W_RSZ] = .DESC[DSC$W_LENGTH];
3314 2 RAB[RAB$L_RBF] = .DESC[DSC$A_POINTER];
3315 2
3316 3 IF NOT (STATUS = $PUT(RAB=RAB))
3317 2 THEN
3318 2 SIGNAL_STOP(QUEMAN$WRITEERR,1,OUTPUT_DESC,.STATUS,.RAB[RAB$L_STV]);
3319 2
3320 1 END;
```

.EXTRN SYSSPUT

0004 00000 PUT_OUTPUT:

	52	0000'	CF	9E	00002	.WORD	Save R2
	50	04	AC	D0	00007	MOVAB	RAB+34, R2
	62		60	B0	0000B	MOVL	DESC, R0
06	A2	04	A0	D0	0000E	MOVW	(R0), RAB+34
		DE	A2	9F	00013	MOVL	4(R0), RAB+40
00000000G	00		01	FB	00016	PUSHAB	RAB
17			50	E8	0001D	CALLS	#1, SYSSPUT
						BLBS	STATUS, 1\$

```
: 3256
:
: 3291
:
: 3292
: 3294
:
```

QUEMANSHO
V04-000

SHOW QUEUE command

H 9
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 111
(23)

	EA	A2	DD	00020	PUSHL	RAB+12
		50	DD	00023	PUSHL	STATUS
	22	A2	9F	00025	PUSHAB	OUTPUT_DESC
		01	DD	00028	PUSHL	#1
		8F	DD	0002A	PUSHL	#16453844
00000000G 00		05	FB	00030	CALLS	#5, LIB\$STOP
		04	00037	1\$:	RET	

: 3296

: 3298

; Routine Size: 56 bytes, Routine Base: CODE + 25AC


```

3322 3299 1 ROUTINE EXIT_HANDLER: NOVALUE=
3323 3300 1
3324 3301 1 ++
3325 3302 1
3326 3303 1 FUNCTIONAL DESCRIPTION:
3327 3304 1 This routine is an exit handler that is established as long as the
3328 3305 1 process owns open wildcarding. Its function is to cancel $GETQUI
3329 3306 1 wildcarding if an image exit should occur.
3330 3307 1
3331 3308 1 INPUT PARAMETERS:
3332 3309 1 Standard exit handler parameters (not used).
3333 3310 1
3334 3311 1 IMPLICIT INPUTS:
3335 3312 1 NONE
3336 3313 1
3337 3314 1 OUTPUT PARAMETERS:
3338 3315 1 NONE
3339 3316 1
3340 3317 1 IMPLICIT OUTPUTS:
3341 3318 1 NONE
3342 3319 1
3343 3320 1 ROUTINE VALUE:
3344 3321 1 NONE
3345 3322 1
3346 3323 1 SIDE EFFECTS:
3347 3324 1 NONE
3348 3325 1
3349 3326 1 --
3350 3327 1
3351 3328 2 BEGIN
3352 3329 2 $GETQUI(FUNC=QUI$_CANCEL_OPERATION);
3353 3330 1 END;

```

.EXTRN SYSS\$GETQUI

```

0000 00000 EXIT_HANDLER:
7E 7C 00002 .WORD Save nothing
7E 7C 00004 CLRQ -(SP)
01 7D 00006 CLRQ -(SP)
7E D4 00009 MOVQ #1, -(SP)
07 FB 0000B CLRL -(SP)
04 00012 CALLS #7, SYSS$GETQUI
RET

```

```

: 3299
: 3329
:
:
:
: 3330

```

; Routine Size: 19 bytes, Routine Base: CODE + 25E4

QUEMANSHO
V04-000

SHOW QUEUE command

J 9
16-Sep-1984 00:18:05
14-Sep-1984 12:08:45

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]QUEMANSHO.B32;1

Page 113
(25)

: 3355
: 3356
3331 1 END
3332 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
DATA	663	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	9719	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
. ABS .	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	289	2	581	00:01.0

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/LIS=LIS\$:QUEMANSHO/OBJ=OBJ\$:QUEMANSHO MSRC\$:QUEMANSHO/UPDATE=(ENH\$:QUEMANSHO)

: Size: 7223 code + 3159 data bytes
: Run Time: 02:21.7
: Elapsed Time: 07:35.9
: Lines/CPU Min: 1411
: Lexemes/CPU-Min: 21065
: Memory Used: 632 pages
: Compilation Complete

0050 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PRONOUNCE
LIS

QUEMAN
LIS

MATCHKEY
LIS

PUTCLMSG
LIS

QUEMANMSG
LIS

PASSWORDS
LIS

QUEMANSHO
LIS

0051 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY